Titre: Title:	Rapid restoration of tomographical images
Auteur: Author:	Shelby Pereira
Date:	1997
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Pereira, S. (1997). Rapid restoration of tomographical images [Master's thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/6731/

UNIVERSITÉ DE MONTRÉAL

Rapid Restoration of Tomographical Images

Shelby Pereira

Institut de Génie Biomédical

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION

DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES

Génie Biomédical

Octobre 1997

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE

Ce mémoire intitulé:

Rapid Restoration of Tomographical Images

présenté par: Shelby Pereira

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

Bertrand, Michel, Ph.D., président

Goussard, Yves, Ph.D., membre et directeur de recherche

Lecomte, Roger, Ph.D., membre

Acknowledgements

I would first like to thank my director Professor Yves Goussard who proposed the image restoration project and supervised it, providing many suggestions and helpful comments along the way. I would also like to thank my director for the opportunity to do a three month exchange in France. Thanks to Jérome Idier for supervising the project work in France and Frédérique Champagnat for his help in the programming work done in France. I also thank Professor Roger Lecomte and Professor Michel Bertrand for agreeing to be part of the jury during my defense and for evaluating the work on very short notice. Finally, many thanks to the various students which I worked with in Montreal and Paris and in particular Marc Fayolle and Nicolas Villain for their help in clarifying many points related to Markov Random Fields.

Résumé

Dans cette thèse, on développe des méthodes de régularisation pour restaurer des images. Le travail s'insère dans un projet global dont le but est de développer une prothèse personnalisée du genou. La restauration d'images est la première étape d'une série de traitements qui nous permet d'obtenir la surface du genou à partir des images fournies par un tomographe.

De telles images sont floues et bruitées. On modélise cette dégradation par une convolution entre une PSF (Point Spread Function) et l'image originale à laquelle s'ajoute un bruit. On observe que si l'on estime l'image originale à partir de l'image observée avec des méthodes simples comme les moindres carrés, les solutions obtenues sont instables ; ceci traduit le caractère mal-posé du problème. Par conséquent on utilise une approche de régularisation où de l'information *a priori* est introduite dans le système, ce qui conduit à des solutions stables ; cependant, un compromis doit être effectué entre l'information *a priori* et l'information contenue dans l'image observée.

Deux approches sont analysées : le filtrage de Kalman et le filtrage de Wiener. Ces deux filtres font des hypothèses similaires sur le système. Le filtre de Kalman ordinaire est un filtre récurrent qui, dans notre cas, présente un volume de calcul élevé. Pour limiter cet inconvénient, on développe des méthodes de factorisation qui réduisent le volume de calcul et améliorent la stabilité numérique. On développe aussi un filtre de Wiener contraint qui est inspiré du filtre de Wiener classique. Cependant,

ce filtre utilise un support limité et l'opération de filtrage se réduit à une convolution entre un filtre spatialement invariant l'image observée. Ce filtre s'étend facilement à trois dimensions où la corrélation entre tranches voisines peut être exploitée. De plus, il est aussi possible d'effectuer un filtrage adaptatif dans lequel les caractéristiques du filtre varient spatialement afin de lisser les régions homogènes et restaurer les frontières.

La principale contribution originale de cette thèse est le développement d'un algorithme basé sur la technique de validation croisée pour estimer le paramètre de réglage du filtre de Wiener et ainsi rendre la méthode de restauration *non-supervisée*.

On observe que les méthodes basées sur le filtre de Kalman ainsi que le filtre de Wiener fournissent des résultats presque parfaits quand le rapport signal à bruit est supérieur à 20dB. Pour des rapports signal à bruit plus bas, le filtre de Wiener adaptatif conduit à des meilleurs résultats que le filtre de Kalman et la qualité du résultat est acceptable jusqu'à un rapport signal à bruit de 10dB.

Abstract

In this thesis, we deal with regularization methods as applied to the problem of image restoration within the global context of a project whose ultimate goal is the construction of a knee prosthesis. Image restoration is one phase of a series of image processing algorithms which allows us to reconstruct the surface of the knee from the slices obtained from a tomographical scanner.

The slices obtained from the scanner are blurred and noisy. We model the degradation by a convolution between a point spread function (impulse response in 1-D) and the original image with an additive noise. We find that in attempting to estimate the original image from the observed image, simplistic methods, such as the least squares method, yield unstable solutions because the problem is ill-posed. Consequently we adopt a regularization approach in which *a priori* knowledge is introduced into the system. The introduction of *a priori* information leads to a stable solution in which a compromise is made between the *a priori* information and the observed image.

Two approaches are considered: Kalman filtering and Wiener filtering. Both filters make similar hypotheses about the system. The Kalman filter is a recursive filter which in its basic form suffers from a high computational load. Consequently we develop factorization methods which reduce the load and improve the stability. We also develop the constrained Wiener filter in which is inspired by the classical Wiener filter. However, in this case the filter support is constrained and the image is restored

by convolving the filter with the observed image. This filter is easily extended to 3D where the correlation between neighboring slices can be exploited. Furthermore it is also possible to filter images adaptively. That is the filter can be changed locally in order to smooth homogeneous regions while sharpening edges.

The central original contribution of this thesis is the development of an algorithm based on Cross Validation which estimates the signal to noise ratio for the Wiener filter. This allows un-supervised restoration to be performed.

We find that the Kalman based filters and the Wiener filter yield restored images of near perfect quality when the SNR is 20dB or higher. At lower SNRs the use of an adaptive Wiener filter yields a superior result to the Kalman filter and the quality of the result is acceptable down to an SNR of 10 dB.

Condensé

Le travail décrit dans ce mémoire se situe dans le domaine de la restauration d'images. Il s'insère dans un projet global dont le but est de développer une prothèse personnalisée du genou. Une telle prothèse nécessite d'obtenir la géométrie du contour de l'os avec une précision de l'ordre du millimètre à partir d'images tomographiques. En raison de la difficulté d'estimer un modèle géométrique de l'articulation à partir des données brutes recueillis par le tomographe, le point de départ est constitué par les coupes déjà reconstruites et le problème est divisé en quatre étapes plus simples: restauration, segmentation, extraction des contours, et reconstruction 3-D de la surface. L'objectif de la restauration est de réduire le bruit et le flou; ensuite, lors de la segmentation, on amène l'image à deux niveaux qui représentent respectivement l'os et le tissu. Par la suite, on extrait les contours et on construit une surface 3D lisse en utilisant une technique de krigeage. Dans ce qui suit on examine en détail l'étape de restauration d'images.

Objectifs

On peut citer 5 objectifs principaux qu'un algorithme de restauration devrait atteindre. L'importance de ces objectifs varie selon l'application. Les objectifs sont:

1. La suppression du flou
2. L'élimination du bruit
3. La rapidité du traitement

4. La possibilité de traiter des données en 3-D

5. La possibilité d'effectuer le traitement de manière non-supervisée.

Dans presque tous les cas, les deux premiers objectifs sont les plus importants puisqu'ils influencent directement la précision et, en général, on doit chercher un compromis acceptable entre ceux-ci. Les images tomographiques contiennent du flou qui est produit par les algorithmes de reconstruction utilisés pour produire des images à partir des données brutes et par les imperfections du tomographe. Les images contiennent aussi du bruit qui vient de la précision numérique finie de l'ordinateur ainsi que des erreurs dans les mesures. Aux frontières on cherche à réduire le flou pour augmenter la précision quand on extrait les contours. Dans les régions homogènes, par contre, on veut réduire le bruit à un niveau où il ne gène plus l'interprétation de l'image. Par conséquent on s'intéressera en particulier aux filtres qui peuvent s'adapter localement selon que l'on traite une région homogène ou une transition.

La rapidité est aussi importante dans notre cas parce qu'en général entre 30 et 40 tranches doivent être traitées pour obtenir la géométrie de la surface du genou. Chaque tranche est une image de 200x200 pixels. Ceci représente 1.5 million de pixels à traiter. Il est donc nécessaire d'avoir des algorithmes assez rapides pour effectuer en un temps raisonnable le traitement. Les données étant des tranches d'une surface 3-D, il est désirable que l'algorithme de restauration profite de la corrélation entre les tranches pour effectuer la restauration puisque les tranches adjacentes sont très similaires, d'où la nécessité d'un traitement 3D.

Enfin, souvent, les algorithmes ne sont pas utilisés par les personnes qui les ont développés. C'est le cas ici parce que les algorithmes seront utilisés par des personnes du domaine médical. Par conséquent, il est souhaitable que les algorithmes fassent une sélection automatique des paramètres qui règlent le compromis entre la réduction du flou et la réduction du bruit.

Problématique :

Pour obtenir un modèle simple, on fait l'hypothèse que, en première approximation, les distorsions provenant du tomographe sont linéaires et invariantes et que le bruit est additif. Cependant, on observe en pratique que l'étendue de la PSF (Point Spread Function (réponse impulsionnelle)), qui se modélise bien comme Gaussienne, varie d'environ 20% entre le centre et les bords de l'image. Cependant, nos expériences indiquent qu'il y a peu à gagner en tenant compte de cette variation. On utilise alors un modèle linéaire invariant puisque cela conduit à des solutions explicites et des algorithmes moins complexes. De plus, ce modèle dans le cas des images tomographiques est suffisamment bon pour donner des résultats intéressants. Le modèle pour un tel système est le suivant :

$$y = Hx + b \tag{1}$$

où y est l'image observée. H est construit avec les échantillons de la PSF qui modélise les distorsions du tomographe. b est le bruit et x est l'image à estimer. Dans notre cas qui utilise une PSF invariante, la matrice H prend la forme Toeplitz-block-Toeplitz.

Le problème de restauration d'images rentre dans la classe des problèmes inverses. Dans ces problèmes on cherche à estimer un objet qui a subit des distorsions et est noyé dans un bruit. La manière la plus immédiate de résoudre l'Équation 1 est d'utiliser le maximum de vraisemblance, les moindres carrés ou l'inversion généralisée. Cependant, ces solutions sont instables car une petite variation dans l'image observée peut conduire à une grande variation dans l'image estimée.

Une approche qui conduit à des solutions stables est l'approche de régularisation. L'idée principale de cette approche consiste à introduire de l'information *a priori* dans le système et ensuite de chercher un compromis acceptable entre l'*a priori* et les

observations. Pour ce faire on définit la solution comme :

$$\hat{\boldsymbol{x}}(\mu, \boldsymbol{y}) = \arg\min_{\boldsymbol{x}} \{J_1(\boldsymbol{x}, \hat{\boldsymbol{x}}_0) + \mu J_2(\boldsymbol{x}, \hat{\boldsymbol{x}}_\infty)\} \tag{2}$$

où J_1 exprime la fidélité à $\boldsymbol{x}_0$, donc aux données, et J_2 la proximité de $\boldsymbol{x}_\infty$ qui représente l'*a priori*. $\hat{\boldsymbol{x}}_0$ est la solution du problème sans régularisation, $\boldsymbol{x}_\infty$ est une image douce et μ est un paramètre qui pondère l'importance des deux fonctions. Le choix des fonctions J_1 et J_2 influence fortement la qualité de la solution ainsi que la complexité des algorithmes utilisés pour minimiser le critère résultant.

On peut séparer les méthodes existantes en deux grandes classes : celles des critères quadratiques et celle des critères non-quadratiques. La classe "quadratique" conduit à des solutions où les frontières sont lissées, mais les algorithmes qui minimisent le critère sont en général rapides. La classe "non-quadratique" inclut entre autres les méthodes basées sur les champs de Markov qui permettent de modéliser des régions homogènes séparées par des discontinuités. Cependant les critères résultants sont plus difficile à minimiser et les algorithmes sont plus compliqués. Ici on se limite au cas quadratique car la rapidité et le volume de calcul sont des paramètres importants dans notre application. On veut aussi pouvoir régler localement le compromis entre le lissage et la restauration. On s'intéressera alors en particulier aux filtres capables d'effectuer une restauration adaptative. On vérifiera en pratique que minimiser un critère quadratique sur nos images est suffisant pour obtenir des résultats adéquats.

Il est souvent pratique de formuler le problème dans un cadre stochastique. Pour faire ceci on modélise notre image à estimer $\boldsymbol{x}$ par une distribution $p(\boldsymbol{x})$. Cette distribution contient notre *a priori*. L'utilisation d'une distribution est intéressante puisqu'elle nous permet de modéliser le fait que nous nous intéressons à une large

classe d'images et que par conséquent il y a des incertitudes dans notre *a priori*. Nous spécifions aussi la distribution $p(\boldsymbol{y}|\boldsymbol{x})$ qui traduit l'information contenue dans les observations. On combine ces deux probabilités en utilisant la règle de Bayes pour obtenir la probabilité *a posteriori*:

$$p(\boldsymbol{x}|\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})/p(\boldsymbol{y}) \tag{3}$$

où $p(\boldsymbol{y})$ sert à normaliser la distribution et $p(\boldsymbol{x}|\boldsymbol{y})$ résume toute l'information sur l'image inconnue une fois que les mesures sont effectuées. Il s'agit maintenant de choisir une solution. Une possibilité est le *maximum a posteriori* MAP défini par:

$$\hat{\boldsymbol{x}} = \boldsymbol{x}_{MAP} = \arg\max p(\boldsymbol{x}|\boldsymbol{y}) \tag{4}$$

Le MAP est fréquemment utilisé parce qu'il est facile à calculer. Si on choisit les distributions définies ci-dessous:

$$p(\boldsymbol{y}|\boldsymbol{x}) \propto \exp -\frac{1}{2}\boldsymbol{J}_1(\boldsymbol{x}, \hat{\boldsymbol{x}}_0) \tag{5}$$

$$p(\boldsymbol{x}) \propto \exp -\frac{\mu}{2}\boldsymbol{J}_2(\boldsymbol{x}, \hat{\boldsymbol{x}}_\infty) \tag{6}$$

alors la solution MAP est identique à celle de l'équation 2. On voit ainsi que l'approche déterministe et l'approche stochastique sont deux interprétations d'une même technique.

Dans le cas quadratique $\boldsymbol{J}_1$ et $\boldsymbol{J}_2$ prennent la forme qui suit:

$$J_1 = \|\boldsymbol{x} - \hat{\boldsymbol{x}}_0\|^2_{\boldsymbol{H}^t\boldsymbol{H}} \tag{7}$$

$$J_2 = \|\boldsymbol{x} - \hat{\boldsymbol{x}}_\infty\|^2_{\boldsymbol{P}^t\boldsymbol{P}} \tag{8}$$

où $\boldsymbol{P}$ est une matrice symétrique, définie positive. Dans le cadre stochastique, ceci correspond à introduire des distributions Gausiennes. On choisira ainsi la matrice $\boldsymbol{P}^t\boldsymbol{P}$ comme l'identité ou une matrice définie positive dans le cas ou on veut introduire de la corrélation. L'introduction de la corrélation est logique puisque, les images étant assez lisses en général, les voisins d'un pixel nous apportent une information importante sur ce pixel.

Minimiser l'équation 2 nous donne :

$$\hat{\boldsymbol{x}}(\mu, \boldsymbol{y}) = (\boldsymbol{H}^t\boldsymbol{H} + \mu\boldsymbol{P}^t\boldsymbol{P})^{-1}\boldsymbol{H}^t\boldsymbol{y} \tag{9}$$

où nous avons choisi $\boldsymbol{x}_\infty$ comme image constante égale à 0 partout. L'équation 9 décrit une famille de solutions qui dépend du paramètre μ. Le problème avec cette équation est que dans notre cas la matrice $(\boldsymbol{H}^t\boldsymbol{H} + \mu\boldsymbol{P}^t\boldsymbol{P})$ est de dimension $(MM \times NN)$ où M et N sont les dimensions de l'image. Une image 200×200 nous donne alors une matrice de taille 40000×40000 à inverser ce qui est trop coûteux en temps de calcul.

Pour développer une solution calculée récursivement, on adopte l'interprétation stochastique. $p(\boldsymbol{x})$ et $p(\boldsymbol{y}|\boldsymbol{x})$ correspondant à 7 et 8 sont alors définies par:

$$p(\boldsymbol{y}|\boldsymbol{x}) \propto e^{\frac{-1}{2\sigma_n^2}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|^2} \tag{10}$$

$$p(\boldsymbol{x}) \propto e^{\frac{-1}{2\sigma_x^2}\boldsymbol{x}^t\boldsymbol{P}^{-1}\boldsymbol{x}} \tag{11}$$

où :

$$\mu = \frac{\sigma_n^2}{\sigma_x^2} \tag{12}$$

$$\boldsymbol{P} = (\boldsymbol{Q}^t\boldsymbol{Q})^{-1} \tag{13}$$

La solution de ce système est identique à celle de l'Équation 9 et peut être calculée par le filtre de Kalman qui suit.

$$\hat{\boldsymbol{x}}_{k+1|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{k}_k e_k^{-1}[\boldsymbol{y}_{k-1} - \boldsymbol{H}_k \hat{\boldsymbol{x}}_{k\,k-1}] \quad (14)$$

$$e_k = \boldsymbol{h}_k^t \boldsymbol{P}_{k|k-1} \boldsymbol{h}_k + r_k \quad (15)$$

$$\boldsymbol{k}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{h}_k \quad (16)$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{k}_k e_k^{-1} \boldsymbol{k}_k^t \quad (17)$$

avec les initialisations:

$$\boldsymbol{P}_{0|-1} = \text{identité ou Toeplitz-block-Toeplitz (avec corrélation)}$$

$$\hat{\boldsymbol{x}}_{0|-1} = \boldsymbol{E}[x]$$

Cependant, avec ce filtre de Kalman, nous avons des problèmes de volume de calcul puisque la matrice $\boldsymbol{P}$ est aussi une matrice 40000×40000 qu'on doit mettre à jour à chaque récurrence.

Pour réduire le volume de calcul, au lieu de propager la matrice $\boldsymbol{P}$ on propage une matrice $\delta \boldsymbol{P}_k = \boldsymbol{P}_k - \boldsymbol{P}_{k-1}$. Si la matrice $\boldsymbol{P}$ est égale à l'identité ou à une structure Toeplitz, l'incrément $\delta \boldsymbol{P}_k$ peut se factoriser comme:

$$\delta \boldsymbol{P}_k = \boldsymbol{Y}_k \boldsymbol{M}_k \boldsymbol{Y}_k \quad (18)$$

où la matrice $\boldsymbol{M}$ est 3×3. Avec cette factorisation on peut propager les matrices $\boldsymbol{Y}$ et $\boldsymbol{M}$ en utilisant les équations de Chandrasekhar. Ceci constitue une grosse réduction dans le volume de calcul, puisque la matrice $\boldsymbol{Y}$ est 40000×3 et $\boldsymbol{M}$ est 3×3.

Un problème avec le filtre de Chandrasekhar est que nous ne pouvons pas faire

du filtrage en 3D à cause de la taille des matrices qui devient beaucoup trop grande même avec la factorisation. On observe également que si on choisit la matrice $\boldsymbol{P}$ Toeplitz (avec corrélation) l'algorithme est numériquement instable. Un autre problème provient du fait que le filtre de Chandrasekhar minimise un critère global et par conséquent, ne permet pas d'adapter le traitement pour lisser les régions homogènes et restaurer les frontières. On propose alors dans ce qui suit une approche par filtrage de Wiener contraint qui nous permet à la fois de traiter de manière adaptative le problème en 3D, et d'introduire de la corrélation sur $\boldsymbol{x}$.

Le filtre de Wiener classique se développe en minimisant le critère d'erreur moyenne quadratique (EMQ) donnée par:

$$\text{EMQ} = \boldsymbol{E}[(\hat{\boldsymbol{x}} - \boldsymbol{x})^2] \tag{19}$$

L'estimée s'écrit alors $\hat{\boldsymbol{x}} = \boldsymbol{W}\boldsymbol{y}$. $\boldsymbol{W}$ étant une matrice égale à $\boldsymbol{R}_{yy}^{-1}\boldsymbol{R}_{xy}$ où $\boldsymbol{R}_{yy}$ et $\boldsymbol{R}_{xy}$ sont des matrices d'autocorrélation et d'intercorrélation qui contiennent les échantillons de $E[\boldsymbol{yy}]$ et $E[\boldsymbol{yx}]$ respectivement. Le problème avec cette approche est la grande taille de la matrice $\boldsymbol{R}_{yy}$ qui nous empêche d'étendre le filtre à trois dimensions. De plus, avec cette approche on ne peut pas modifier localement les propriétés du filtre pour tenir compte de la présence de discontinuités. En particulier, on se rappelle qu'un objectif était de lisser les régions homogènes et restaurer les frontières.

On utilise alors une autre approche qui nous permettra de filtrer l'image en tenant compte des propriétés locales. Pour ce faire, on rend le filtre local en lui imposant une contrainte sur le support. Ceci permet de changer les caractéristiques du filtre en fonction des caractéristiques locales de l'image observée. De plus, un support réduit conduit à une réduction importante dans les volumes de calcul qui permet par la suite

une extension en 3-D. On commence par minimiser le EMQ pour un pixel:

$$E[(\boldsymbol{x}_{i,j} - \hat{\boldsymbol{x}}_{i,j})^2] \tag{20}$$

On cherche une solution avec la forme $\hat{x}_{i,j} = \boldsymbol{w}_{i,j}^t \boldsymbol{y}$ où le support de $\boldsymbol{w}$ est limité à $[s_x^1, s_x^2] \times [s_y^1, s_y^2]$. On désigne $\boldsymbol{w}$ le filtre de Wiener contraint. La minimisation de l'équation 20 nous donne $\boldsymbol{w} = \boldsymbol{R}_{yy}^{-1} \boldsymbol{r}_{xy}$ où $\boldsymbol{R}_{yy}$ et $\boldsymbol{r}_{xy}$ sont cette fois contraints en fonction du support choisi pour $\boldsymbol{w}$. Les échantillons de $\boldsymbol{R}_{yy}$ et $\boldsymbol{r}_{xy}$ sont pris à partir des fonctions de corrélation et intercorrélation :

$$\boldsymbol{r}_{xy}(m,n) = \sum_{k,l} h(k,l) r_{xx}(m+k, n+l) \tag{21}$$

$$\boldsymbol{r}_{yy}(m,n) = \sum_{i,j,k,l} h(k,l)h(i,j) \; r_{xx}(m-k+i, n-l+j) + r_n \delta(k)\delta(l) \tag{22}$$

où h contient les échantillons de la PSF et $\boldsymbol{r}_{xx}$ est la fonction d'autocorrélation de $\boldsymbol{x}$. Maintenant, si on fait l'hypothèse que l'image est stationnaire, on trouve que l'estimé $\hat{\boldsymbol{x}}$ est égal à $\boldsymbol{w} * \boldsymbol{y}$. L'estimé est donc donné par une convolution entre le filtre $\boldsymbol{w}$ et l'image observée. Comme avec le filtre de Wiener contraint est contrôlé par de l'information *a priori* représentée par les quantités $\boldsymbol{r}_{xx}$ et r_n. $\boldsymbol{r}_{xx}$ est équivalent à la matrice de covariance $\boldsymbol{P}$ dans le filtre de Chandrasekhar et le paramètre r_n, qui nous permet d'effectuer un compromis entre la fidélité aux données et l'*a priori*, joue le rôle de μ. On note qu'on peut normaliser $\boldsymbol{r}_{xx}$ sous la forme $r_x \overline{\boldsymbol{r}}_{xx}$ et ensuite exprimer le filtre comme $\boldsymbol{W} = \lambda \overline{\boldsymbol{R}}_{yy}^{-1} \overline{\boldsymbol{r}}_{xy}$ où $\lambda = r_n / r_x$. Ce filtre peut facilement être étendu au cas 3D au prix d'un changement de structure des quantités $\boldsymbol{r}_{xy}$ et $\boldsymbol{R}_{yy}$.

Le filtre de Wiener contraint permet d'effectuer un filtrage adaptatif. On note que le filtre a été établi en minimisant l'EMQ pour un pixel. Par conséquent nous

pouvons utiliser un filtre différent pour chaque pixel. Pour restaurer les frontières et lisser les régions homogènes. nous pouvons utiliser deux filtres: pour lisser les régions homogènes on utilisera un paramètre λ qui est élevé et on choisira un *a priori* avec corrélation. par contre. aux frontières. on prendra un *a priori* sans corrélation et un paramètre λ plus bas. Ceci favorise la restauration. Le choix du filtre local utilisé peut se faire en fonction de la variance locale de l'image observée.

Estimation des paramètres :

Une des contributions originales de ce mémoire est le développement d'un algorithme qui estime le paramètre λ du filtre de Wiener. L'algorithme est basé sur la validation croisée que. pour des raisons de simplicité. nous présentons en 1D. L'idée est d'enlever un point de l'image observée et de l'estimer à partir des autres points dans l'image. L'emploi d'une telle technique est justifié par un théorème de convergence asymptotique selon lequel. à nombre de pixels infini. le paramètre de régularisation qui minimise le critère de validation croisée (VC) conduit à une estimée de $\boldsymbol{x}$ telle que

$$\|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{H}\hat{\boldsymbol{x}}(\mu, \boldsymbol{y})\|^2 \tag{23}$$

est minimale. Mathématiquement. la VC consiste à minimiser le critère:

$$VC(\lambda) = \sum_k (y_k - \hat{y}_k^{-k})^2 \tag{24}$$

où λ est le paramètre qu'on estime. $\hat{y}_k^{-k}$ est la valeur de y_k estimée avec tous les échantillons sauf l'échantillon k. Cette dernière quantité dépend du paramètre λ. On choisit le λ qui minimise ce critère.

Pour calculer les échantillons y_k^{-k} on commence par calculer les échantillons x_k^{-k}

où x_k^{-k} est le $k$ième pixel de l'estimée de $\boldsymbol{x}$ calculée sans l'observation y_k.

$$\hat{\boldsymbol{x}}^{-k} = \tilde{\boldsymbol{W}}\boldsymbol{y}^{-k} \tag{25}$$

ou $\boldsymbol{y}^{-k}$ contient les observations avec le $k$ième échantillon mis a 0. Nous avons montré que $\tilde{\boldsymbol{W}}$ est une matrice dont chaque ligne contient des coefficients d'un filtre de Wiener contraint développé comme dans la section précédente, l'échantillon k étant enlevé du support. $\hat{y}_k^{-k}$ est alors égal à $h^t\tilde{\boldsymbol{W}}\boldsymbol{y}^{-k}$ et on observe que la quantité $h^t\tilde{\boldsymbol{W}}$ est invariante par décalage. On obtient donc tous les échantillons de y_k^{-k} par convolution de $\boldsymbol{h}^t\tilde{\boldsymbol{W}}$ et $\boldsymbol{y}$. Pour calculer le critère VC il suffit maintenant de prendre la norme de la différence entre $\boldsymbol{y}$ et $\boldsymbol{y}_k^{-k}$. La valeur de λ choisie est celle qui minimise le critère et on la désigne λ_{VC}. Le critère ne pouvant pas être minimisé analytiquement, on l'évalue pour plusieurs valeurs de λ et on retient celle qui correspond au minimum. En pratique on ne trouve qu'un seul minimum.

Résultats:

Nos résultats montrent que le filtre de Wiener contraint est celui qui produit la meilleure performance des filtres étudiés. Ce filtre a plusieurs avantages sur les filtres basés sur le filtre de Kalman. En particulier, il présente un volume de calcul bas, peut être mis en oeuvre en 3-D et se prête aux traitements adaptatifs. Ce filtre produit des résultats adéquats jusqu'à un SNR de 10dB ce qui est suffisant en général pour les images tomographiques. Enfin, ce filtre peut être utilisé de manière supervisée grâce à la méthode VC. En pratique on observe que la méthode VC sous-estime le paramètre λ. Pour corriger cet effet de manière empirique, on multiplie le paramètre par un facteur compris entre 2 et 5. Dans le cas adaptatif on utilise un paramètre de 100 fois la valeur de λ_{VC} dans les régions où on veut lisser l'image et on utilise un paramètre de 2 fois la valeur de λ_{VC} sur les transitions. Ceci conduit à un résultat

qui est nettement supérieur à l'approche par filtrage de Kalman. Sur des images synthétiques nous arrivons à restaurer les images presque parfaitement avec le filtrage adaptatif pour un SNR supérieur à 15dB.

Sur des vraies images on trouve que le filtre réussit bien à restaurer les frontières et lisser les régions homogènes. Cependant, il est impossible d'évaluer la performance en terme d'erreur parce que l'image originale n'est pas connue. Donc, il reste à effectuer une étape de validation sur des objets réels dont on connait précisément les dimensions.

Table of Contents

List of Tables

List of Figures

Introduction

The field of image restoration originated in the 1950's and 1960's when satellite images obtained through the U.S. and Soviet Union space programs required the use of image restoration techniques to remove the degradations associated with suboptimal imaging environments. Since this time, image restoration has found application in an increasing number of fields including biomedical imaging. Two specific examples in this domain are magnetic resonance imaging (MRI) (Soltanian-Zodiah, Windham and Yagle 1995) and quantitative autoradiography (QAR) (Goyette, Lain and Kang 1994). Both these imaging techniques produce images which contain noise which can be reduced by applying digital image restoration techniques. In this study, the problem of restoration of X-ray computed tomographical images is considered.

The image restoration techniques presented in this document are used in conjuction with other image processing methods in a project which ultimately involves the construction of a personalized knee prosthesis. Knee prostheses are currently primarily being used in the treatment of arthritis. Arthritis is a disease which affects the cartilage. When the cartilage of the knee deteriorates as the result of arthritis, it is possible to replace it with a knee prosthesis. Two general approaches to designing prostheses can be stated. The first approach, which is the one currently being used is based on the prosthesis of the knee shown on the left hand side of Figure 1. With this approach, the knee is cut to fit the prosthesis.

The problems with the current prostheses include: the necessity of cutting the bone to fit the prosthesis which is a serious surgical operation, and the difficulties involved in making revisions later since the bone may have to be re-cut to fit the prosthesis. Only a very small number of revisions can be made since the bone stock is limited. For this reason, this type of prothesis is rarely used with patients less than 55 years old.

In order to avoid such difficulties and extend the range of prescription of knee prostheses, we are currently developing a second approach. In this approach, the knee prosthesis is designed to fit the bone, essentially to replace the deteriorated cartilage, so that very little or no cutting of the bone is necessary (Figure 1, right side). The prosthesis must approximate the shape of the knee to sub-millimeter accuracy in order to obtain appropriate bonding between the bone tissue and the prosthesis, and in order to limit strain that may induce fatigue and rupture of the prosthesis.

The development of a personalized knee prosthesis can be divided into two stages: the image processing stage and the fabrication stage. In the first stage, X-ray tomography is used to obtain images of slices of the knee. These must be processed to obtain the three dimensional knee geometry since only a finite number of slices are available and furthermore the raw images are too inaccurate to be used directly. In a second stage, the processed data is used to fabricate a prosthesis adapted to the shape of the knee. Since the prosthesis is adapted, the cutting of the bone is significantly reduced. It is expected that the personalized knee prothesis will have a longer life span since it is closely adapted to the shape of the knee.

Here we are interested with the image processing stage. One of the difficulties is to obtain the required sub-millimeter geometric accuracy from the series of CT scan images. Reconstructing the surface of the knee from the raw projection data obtained

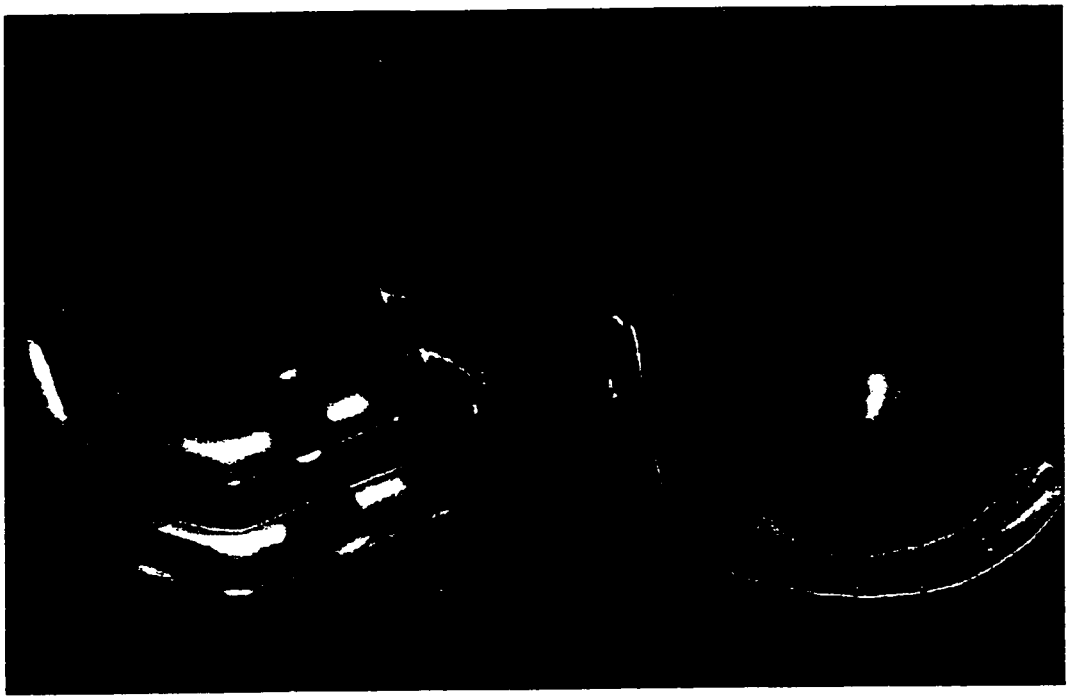


Figure 1: Old and new knee prostheses

from the CT scanner with the required accuracy poses a complex estimation problem. Consequently, the starting point consists of the reconstructed slices provided by the CT scanner, and the task is broken into four simpler components which are treated in succession. These four components are: image restoration, image segmentation, contour extraction and smoothing, and 3-D geometric modeling. In the image restoration phase, blur and noise, present in the raw image, are removed as much as possible. This is followed by image segmentation which consists of separating the image into distinct regions associated with the presence or absence of bone. Contours are then extracted and smoothed. Finally a 3-D interpolation is performed which allows any intermediate slice to be extracted. This is necessary since the original information is contained in discrete slices. Once the image processing stage is complete, the information can then be used to design a prosthesis which conforms to the shape of the knee.

In what follows, we will concern ourselves with the image restoration phase. During this phase, we attempt to remove the distortions caused by the CT-scanner and the reconstruction algorithms used to transform the raw projection data into images. These distortions can be modeled as a spatially varying blur with additive noise. The central task therefore is to reduce this blur and noise as much as possible. However, in attempting to address this problem, we must restrict ourselves to algorithms which process the data efficiently since approximately 40 images, each 200×200 pixels must be processed. We are also interested in algorithms which can automatically select parameters which control the quality of the restoration. This is important as ultimately the algorithms will be used by people who are unfamiliar with the details of the processing technique. Finally we are interested in algorithms which can process the data in three dimensions, since the surface of the knee is a three dimensional object.

In this dissertation, several image restoration algorithms will be presented which attempt to retore images with sufficient precision and speed so as to be viable in the context of the construction of a knee prosthesis. We begin by presenting a survey of the various algorithms available for the general image restoration. Then, these are critiqued with respect to the requirements of our specific project. We will see that many algorithms available for the general image restoration problem can be improved upon in the context of our problem. In particular, we find that with existing algorithms, problems occur when we attempt to extend to three dimensions. One problem is that the volume of computations becomes overwhelming. Another problem is that for a number of algorithms, parameters which control the quality of the image must be precisely set to produce good results and often parameter estimation algorithms do not exist.

In the main section of this thesis, we develop algorithms which attempt to reduce

the shortcomings of the algorithms presented in the literature survey. In Chapter 2 algorithms based on the Kalman filter are developed, but these suffer from the drawback that they cannot be extended to three dimensions due to the large volume of computations required. In Chapter 3, which contains an article, we develop the constrained Wiener filter which is implementable in three dimension. Furthermore, we develop a parameter estimation algorithm which allows the processing to take place in an un-supervised manner. In Chapter 4, the results are presented followed by a conclusion.

Chapter 1

Literature Survey

The images obtained from several fields such as satellite, ultrasound and tomographical imaging contain blur and noise which can be reduced by applying one of a variety of image restoration techniques. This section presents the objectives that any successful image restoration algorithm must obtain, a mathematical formulation of the problem and several restoration algorithms which have been documented in recent scientific publications.

1.1 Problem Formulation and Objectives

1.1.1 Problem Formulation

As was stated earlier many imaging problems consist of removing the blur and noise from the observed image. In this thesis we consider the problem of reducing the blur associated with the CT-scanner and the reconstruction algorithms used to transform the raw data into images. By making measurements of the PSF we find that the blur can be assumed shift-invariant in a first approximation. We will examine to what extent this approximation is valid in the article on Wiener filtering. We also

assume that the noise is additive. While in practice multiplicative noise may exist, it is usually possible to transform the system by using logarithms such that we work with additive noise. Consequently, the observed image is obtained as a convolution between the PSF and the image to be estimated with an additive noise component. The mathematical model is expressed as follows:

$$\boldsymbol{y}(x,y) = \boldsymbol{h}(x,y) * \boldsymbol{x}(x,y) + \boldsymbol{n}(x,y) \tag{1.1}$$

where y is the observed image, h is the PSF, x is the image to be estimated, and n is the noise. In this thesis we will be interested in the problem of estimating x in Equation 1.1. We first define our objectives.

1.1.2 Objectives

Five major objectives can be defined for the general problem of image restoration:

1. Removal of the blur
2. Suppression of the noise
3. Speed
4. Implementation in three dimensions
5. Automatic selection of parameters

The importance of these objectives varies according to the image restoration application. Here we are interested in the restoration of computer tomography (CT) images in the global context of a project whose ultimate goal is the construction of a knee prosthesis. The construction of the prosthesis requires a precise knowledge of the geometry of the surface of the knee. In particular, we require that the error between

the reconstructed surface and the actual knee be under 1mm. Consequently precision in determining the edges of the bone is particularly important. With this in mind, we examine the listed objectives in our context.

Since precision at the edges is of primary importance, it follows that removal of the blur at the edges is our most important objective. Blur exists in tomographical images resulting from the reconstruction algorithms used to transform the raw data into images, the finite number of projections, the size of the transducers, and the imperfections in the system. The blurring effect is particularly noticeable at the edges of objects. In our application it is critical that the blur observed at the edge of the bone be reduced as much as possible as this will lead to better precision in the reconstructed surface.

Another important objective is the suppression of noise in the image to a level where it is possible to accurately recover the contours. In particular, we would like homogeneous regions of the image to appear relatively continuous in the restored image. The noise is present in all observed images due to the imprecise nature of the equipment, artifacts introduced in the reconstruction of the image as well as the finite precision capabilities of the machines used to process the observations. In general, a compromise must be made between reduction of blur and noise suppression. For instance, as will become clear later, attempting to overly suppress noise leads to images without sharp edges. By contrast, if we attempt to restore the boundaries well, we are often left with noise in the resulting image. Consequently we ultimately seek an acceptable compromise between these two objectives.

Speed and the possibility of a three dimensional processing of the image are also important in our application. Given the usual size of our CT images (about 200×200 pixels) and the number of slices (between 30 and 40), we require that the algorithms be efficient in order to process approximately 1.5 million pixels. Furthermore, since

the images vary slowly from one cross-section to the next, we would like to introduce a correlation between the neighboring slices to take this into account. Consequently, algorithms which process the image in three dimensions are desirable. Once again a compromise must be made since sophisticated algorithms for reducing blur and removing noise tend to be costly in computation time. By contrast fast algorithms can be developed which are however less successful in accomplishing the first two objectives. We note that speed and the possibility of three-dimensional processing are not necessary in all applications. For example in satellite imaging, the image restoration is almost exclusively performed in two dimensions.

Finally, many algorithms require a judicious setting of the parameters in order to be successful. In some cases a small change in a parameter leads to a large change in the restored image. We will see that for the restoration algorithms developed in this thesis, a parameter controlling the compromise between suppression of noise and removal of blur must be set. In cases such as ours where the algorithms will ultimately be used commercially by people unfamiliar with the details, it is desirable to have algorithms that automatically choose parameters and can consequently perform the image restoration in an un-supervised manner.

Since ultimately we will need an accurate reconstruction of the surface to construct a prosthesis, removal of the blur and suppression of the noise are the most important stated objectives.

In the literature survey which follows, we look at the most important methods for solving the image restoration which have appeared in the recent scientific publications. The methods we present solve the general image restoration problem as opposed to the problem of restoring CT images. However, these methods will be evaluated with respect to our objectives in order to determine which type of methods best suit our problem.

1.2 Estimators

In this literature survey, we develop several methods for estimating the image from our observation. We then consider algorithms which can be used to calculate the estimate.

Our starting point consists of the generic image restoration problem of Equation 1.1 where the PSF is assumed shift invariant. We begin by putting Equation 1.1 into the following matrix form.

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \tag{1.2}$$

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_0 & 0 & \cdots & \cdots & \cdots \\ \boldsymbol{H}_1 & \boldsymbol{H}_0 & 0 & \cdots & \cdots \\ \vdots & & \ddots & & \cdots \\ \boldsymbol{H}_\gamma & & & \ddots & \\ & \ddots & & & \boldsymbol{H}_0 \\ & & \ddots & & \boldsymbol{H}_1 \\ & & & \ddots & \vdots \\ & & & & \boldsymbol{H}_\gamma \end{bmatrix} ; \boldsymbol{H}_i = \begin{bmatrix} h_{i0} & 0 & \cdots & \cdots & \cdots \\ h_{i1} & h_{i0} & 0 & \cdots & \cdots \\ \vdots & & \ddots & & \cdots \\ h_{i\beta} & & & \ddots & \\ & \ddots & & & h_{i0} \\ & & \ddots & & h_{i1} \\ & & & \ddots & \vdots \\ & & & & h_{i\beta} \end{bmatrix} \tag{1.3}$$

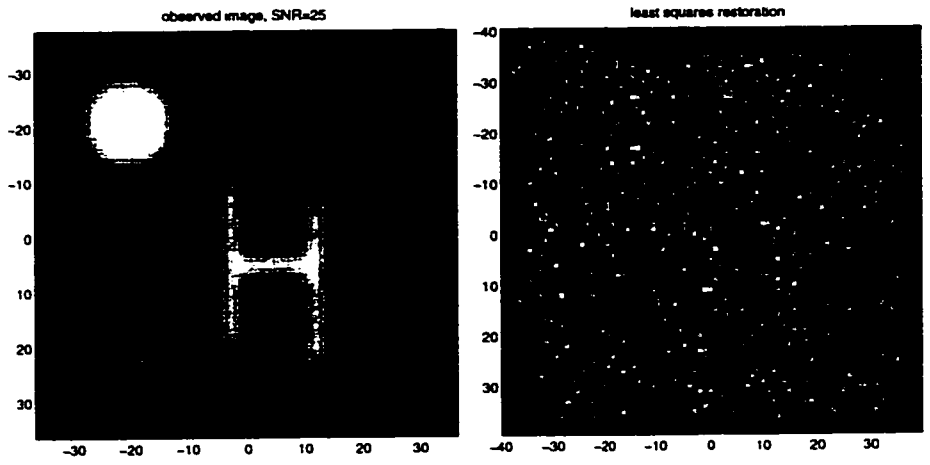
where β and γ are the dimensions of the PSF, and, $\boldsymbol{y}$ and $\boldsymbol{x}$ are vectors containing the images with their samples inserted row by row. The matrix $\boldsymbol{H}$ takes on the special Toeplitz block Toeplitz form which is associated with the convolution with $\boldsymbol{x}$ when the matrix-vector multiplication is performed. We notice that since Equation 1.2 is linear, solving for $\boldsymbol{x}$ by minimizing a quadratic criterion such as the least squares one will yield a solution in explicit form which can be computed by a number of algorithms such as conjugate gradient or Kalman filtering. We also stress that in the image restoration problem, the matrix $\boldsymbol{H}$ is large and yields an enormous computational

problem.

It is well documented in the literature ((Demoment 1989) for example) that simple methods fail to yield suitable solutions for $\boldsymbol{x}$ from the observed $\boldsymbol{y}$ because the matrix $\boldsymbol{H}$ in imaging problems is often ill-conditioned. For example, as illustrated in Figure 1.2, the least squares solution defined by:

$$\hat{\boldsymbol{x}} = (\boldsymbol{H}^t\boldsymbol{H})^{-1}\boldsymbol{H}^t\boldsymbol{y} \tag{1.4}$$

is unacceptable because the small amount of noise present in the observed image has been amplified in the restoration process. The same behavior could have been



(a) Observed image (b) Restored image

Figure 1.1: Example of restoration using the least squares solution

observed with the Moore Penrose solution defined by:

$$\hat{\boldsymbol{x}} = \boldsymbol{H}^{\dagger}\boldsymbol{y} \tag{1.5}$$

The problem with both the Moore Penrose solution and the least squares solution lies in the fact that they are unstable with respect to the noise. In other words, they

have the undesirable property that a small change in the observed image may lead to a large change in the estimated image $\hat{\boldsymbol{x}}$. Consequently the problem of estimating $\boldsymbol{x}$ in Equation 1.2 is termed ill-posed.

Several methods exist of dealing with ill-posed problems. For example, Nashed (Nashed 1981) discusses how acceptable results can be obtained if we search for solutions where:

$$\|\boldsymbol{H}\hat{\boldsymbol{x}} - \boldsymbol{y}\| \leq \epsilon \tag{1.6}$$

where $\epsilon > 0$. Hunt develops iterative methods to solve such problems (Hunt 1977). However one difficulty with iterative methods in the context of ill-posed problems is the numerical instability. In particular, problems with convergence of the iterations are often observed (Hunt 1977). Another method consists of introducing the prior knowledge in the form of constraints which define convex sets. The solution is then obtain by projecting onto the convex sets (Sezan and Stark 1982). The problem here lies in the defining of suitable convex sets. A third family of solutions is the truncated singular value decomposition (TSVD) in which the dimensionality of the solution space is constrained to a subspace in which the eigenvalues of $\boldsymbol{H}$ are greater than some value ϵ. It can be shown that this yields numerical well-conditioned solutions (Demoment 1989, Andrews and Hunt 1977); however a global smoothing of the solution is observed and furthermore the order of the truncation must be arbitrarily specified.

Here we restrict the scope of this literature survey to regularization approaches in which *a priori* knowledge can be injected in an explicit manner into the system in order to produce a stable solution. The *a priori* knowledge can take several forms and in general there is a tradeoff involved between the type of *a priori* knowledge, the

complexity of the resulting problem and the quality of the solution. We now indicate how such a problem can be formulated either in a deterministic framework or in a stochastic framework.

We begin with the deterministic framework. Mathematically, the regularized solution is defined as the solution to the problem:

$$\hat{\boldsymbol{x}}(\mu, \boldsymbol{y}) = \arg\min_{\boldsymbol{x}}\{J_1(\boldsymbol{x}, \hat{\boldsymbol{x}}_0) + \mu J_2(\boldsymbol{x}, \hat{\boldsymbol{x}}_\infty)\} \tag{1.7}$$

where $\boldsymbol{x}_0$ is the solution to the un-regularized problem and $\boldsymbol{x}_\infty$ is an object chosen *a priori* usually selected to be a smooth image. J_1 is the distance between the solution and $\boldsymbol{x}_0$ and expresses fidelity to the observations. J_2 is the distance between the solution and the *a priori* object $\boldsymbol{x}_\infty$ and thus expresses fidelity to the *a priori* knowledge. Finally, μ weights the importance of the two terms. In order to obtain a solution, we must specify these two distance measures, determine a value for the parameter μ and finally develop an algorithm which will give us the minimum. Before discussing various choices of these variables, we demonstrate how the problem can be equivalently formulated stochastically.

We formulate the problem stochastically by adopting a Bayesian approach. In this approach the observed image and the image to be estimated are treated as probability distributions. The image is modeled by the distribution $p(\boldsymbol{x})$ which is chosen *a priori*. Using a probability distribution for the image allows us to model the fact that we are modeling a large class of image and consequently we have uncertainties in our *a priori* knowledge. We also choose the distribution $p(\boldsymbol{y}|\boldsymbol{x})$ *a priori*. This distribution gives us the probability of the observed image given the image to be estimated $\boldsymbol{x}$.

Using Bayes' rule allows us to combine the *a priori* knowledge contained in $p(\boldsymbol{x})$ with the information in the observations contained in $p(\boldsymbol{y}|\boldsymbol{x})$ to obtain the *a posteriori*

distribution

$$p(\boldsymbol{x}|\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})/p(\boldsymbol{y}) \tag{1.8}$$

where $p(\boldsymbol{y})$ is a normalizing constant. Thus, all the information we have about $\boldsymbol{x}$ after the measurements have been made is summarized in the distribution $p(\boldsymbol{x}|\boldsymbol{y})$. The problem now is to obtain a solution from the distribution $p(\boldsymbol{x}|\boldsymbol{y})$.

One solution that solves the problem of selecting an image $\boldsymbol{x}$ given the observed image $\boldsymbol{y}$ is the choice of $\boldsymbol{x}_{MAP}$ defined by

$$\hat{\boldsymbol{x}} = \boldsymbol{x}_{MAP} = \arg\max p(\boldsymbol{x}|\boldsymbol{y}) \tag{1.9}$$

that is we choose the $\boldsymbol{x}$ which maximizes the probability of $\boldsymbol{x}$ given our observations. We might also choose the *a posteriori* mean (Dacunha-Castelle and Duflo 1982), however the MAP estimate is in most cases easier to calculate. We note that if we choose $p(\boldsymbol{x})$ and $p(\boldsymbol{y}|\boldsymbol{x})$ as:

$$p(\boldsymbol{y}|\boldsymbol{x}) \propto \exp -\frac{1}{2}\boldsymbol{J}_1(\boldsymbol{x}, \hat{\boldsymbol{x}}_0) \tag{1.10}$$

$$p(\boldsymbol{x}) \propto \exp -\frac{\mu}{2}\boldsymbol{J}_2(\boldsymbol{x}, \hat{\boldsymbol{x}}_\infty) \tag{1.11}$$

we immediately obtain:

$$p(\boldsymbol{x}|\boldsymbol{y}) = exp-\frac{1}{2}[\boldsymbol{J}_1(\boldsymbol{x}, \hat{\boldsymbol{x}}_0) + \mu\boldsymbol{J}_2(\boldsymbol{x}, \hat{\boldsymbol{x}}_\infty)] \tag{1.12}$$

and it is clear that maximizing $p(\boldsymbol{x}|\boldsymbol{y})$ to yield $\boldsymbol{x}_{MAP}$ is equivalent to minimizing Equation 1.7. Consequently if we choose the probability densities as indicated above, the Bayesian approach and the minimization of the criterion in Equation 1.7 become

equivalent problems.

Having pointed out the equivalence between the stochastic and deterministic frameworks, we now turn our attention to the problem of choosing the function J_1 and J_2 in Equation 1.7 or the equivalent probability densities $p(\boldsymbol{y}|\boldsymbol{x})$ and $p(\boldsymbol{x})$. We can separate the criteria into two general groups depending on the choice of J_1 and J_2, namely quadratic and non-quadratic. As we will see, a quadratic criterion along with a linear model leads to an explicit solution which is relatively easy to calculate by a variety of available algorithms. However the solutions obtained contain transitions which tend to be smooth. By contrast, with non-quadratic criteria, modeling of homogeneous separated by transitions can be done explicitly by choosing J_2 to be, for example, a Markov Random Field (MRF). This will be illustrated further. However, in this case, the solutions are not explicit, and the minimization problem is much more challenging algorithmically.

Quadratic criteria use the following definitions of J_1 and J_2:

$$J_1 = \|\boldsymbol{x} - \hat{\boldsymbol{x}}_0\|^2_{\boldsymbol{H}^t\boldsymbol{H}} \tag{1.13}$$

$$J_2 = \|\boldsymbol{x} - \hat{\boldsymbol{x}}_\infty\|^2_{\boldsymbol{Q}^t\boldsymbol{Q}} \tag{1.14}$$

where $\boldsymbol{Q}$ is a symmetric, positive definite matrix often chosen to be the identity matrix or as a Toeplitz matrix if one wants to introduce correlation in the image. The above choice is often referred to as Tikhonov regularization. In some contexts, including the conjugate gradient algorithm as we will see, it is convenient to formulate the minimization problem with J_1 and J_2 defined as above in the following manner

(Gull 1984):

$$\hat{\boldsymbol{x}} = \min \left\| \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix} - \begin{bmatrix} \boldsymbol{H} \\ \nu \boldsymbol{L} \end{bmatrix} \boldsymbol{x} \right\|_2 \tag{1.15}$$

It can be easily verified that with $\nu = \sqrt{\mu}$ and $\boldsymbol{L} = \boldsymbol{Q}$, Equation 1.15 and Equation 1.7 are equivalent.

In the stochastic framework, the above quadratic problem is equivalent to choosing the probability density functions as:

$$p(\boldsymbol{y}|\boldsymbol{x}) \propto e^{\frac{-1}{2\sigma_n^2}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|^2} \tag{1.16}$$

$$p(\boldsymbol{x}) \propto e^{\frac{-1}{2\sigma_x^2}\boldsymbol{x}^t \boldsymbol{P}^{-1}\boldsymbol{x}} \tag{1.17}$$

where :

$$\mu = \frac{\sigma_n^2}{\sigma_x^2} \tag{1.18}$$

$$\boldsymbol{P} = (\boldsymbol{Q}^t\boldsymbol{Q})^{-1} \tag{1.19}$$

$$\tag{1.20}$$

We now derive the closed-form of the solution. Calculating the gradient with respect to $\boldsymbol{x}$ of Equation 1.7 with quadratic cost functions yields:

$$\hat{\boldsymbol{x}}(\mu, \boldsymbol{y}) = \boldsymbol{x}_\infty + (\boldsymbol{H}^t\boldsymbol{H} + \mu\boldsymbol{Q}^t\boldsymbol{Q})^{-1}\boldsymbol{H}^t(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}_\infty) \tag{1.21}$$

Choosing $\boldsymbol{x}_\infty$, the ultra-smooth object, to be a constant image equal to 0 everywhere yields the regularized least-squares solution:

$$\hat{\boldsymbol{x}}(\mu, \boldsymbol{y}) = (\boldsymbol{H}^t\boldsymbol{H} + \mu\boldsymbol{Q}^t\boldsymbol{Q})^{-1}\boldsymbol{H}^t\boldsymbol{y} \tag{1.22}$$

Equation 1.22 is also referred to as the ridge regression equation which has been studied extensively in the context of spline smoothing problems (Wahba 1985). The matrix in Equation 1.22 to be computed is of the order of the number of pixels of the image and in general it cannot be inverted directly due to its large size. In the literature survey, we will examine the Kalman filter and the conjugate gradient algorithms which compute the solution given by 1.22 in an iterative or recursive manner.

We now turn our attention to the second group of criteria, namely those which are non quadratic. In general J_1 remains quadratic while J_2 is often chosen to be a function which depends on local interactions in the image. In this way, special characteristics such as the presence of homogeneous zones separated by transitions can be introduced into the *a priori*. Choosing J_2 in this way may yield a criterion which is not necessarily convex, and consequently minimization is not always straightforward. In the Bayesian framework, this approach corresponds to adopting a MRF prior model $p(\boldsymbol{x})$ for the image to be restored. We now present this class of models and the corresponding restoration techniques.

A Markov random field is a model in which the probability of certain pixel having a certain intensity given all the other pixels in the image is equal to the conditional probability of that pixel given a set of pixels in its neighborhood. Mathematically we have

$$P(x_{i,j}|\boldsymbol{X} \text{ except } x_{i,j}) = P(x_{i,j}|x_{k,l})k,l \in \mathcal{N}(x_{i,j}) \tag{1.23}$$

where $\boldsymbol{X}$ is all the pixels of the image, $x_{i,j}$ is a given pixel in the image and $\mathcal{N}$ refers to a neighborhood as illustrated in figure 1.2. These local conditional probability distributions must be chosen to be consistent with the global distribution $p(\boldsymbol{x})$. The

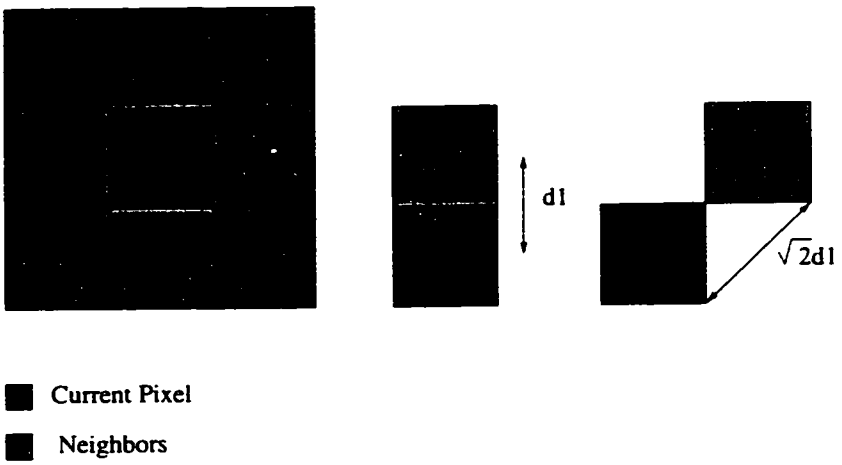


Figure 1.2: Neighbors and inter-pixels distances

Hammersley-Clifford theorem proves that this is possible when $p(\boldsymbol{x})$ is chosen to be a Gibb's distribution. Consequently, we have in this case a Gibb's-Markov equivalence. To adopt this approach in the Bayesian framework we choose $p(\boldsymbol{x})$ as a Gibb's distribution and $\boldsymbol{p(y|x)}$ as a Gaussian as follows:

$$\boldsymbol{p(y|x)} = \exp(-\frac{\|\boldsymbol{y}-\boldsymbol{Hx}\|^2}{2\sigma_n^2}) \tag{1.24}$$

$$\boldsymbol{p(x)} = \boldsymbol{Z}^{-1}\exp(-\frac{1}{T}\sum_{c\in C}\boldsymbol{V}_c(\boldsymbol{x})) \tag{1.25}$$

$$\tag{1.26}$$

where σ^2 is the variance of the noise, c represents a clique of neighboring pixels in a collection C,that is a set of pixels such that any two elements in the set are neighbors of each other, $\boldsymbol{V}_c(x)$ is a penalty assessed for local inter-pixel behavior, Z is the partition function used to normalize the probability distribution and T is a constant known as the "temperature" (Derin and Kelly 1989). It now follows that computing $\hat{\boldsymbol{x}}_{MAP}$ in the Bayesian framework is equivalent to minimizing the cost function defined

by:

$$J(x) = \frac{\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2}{2\sigma_n^2} + \lambda \sum_{c \in C} \boldsymbol{V}_c(\boldsymbol{x}) \tag{1.27}$$

where the parameter λ replaces T. The link to the deterministic framework now becomes clear since in 1.7, we simply choose J_1 to be quadratic as before and J_2 to be $\sum_{c \in C} \boldsymbol{V}_c(\boldsymbol{x})$ with $\mu = \sigma_n^2 \lambda / 2$ to recover 1.27. We also observe that by choosing V_c to be quadratic we recover the quadratic criterion whose solution is given by Equation 1.7 with $\boldsymbol{Q}$ chosen to be a difference operator such that $\boldsymbol{x}^t \boldsymbol{Q}^t \boldsymbol{Q} \boldsymbol{x}$ gives an inter pixel quadratic penalty over the clique. This implies that the quadratic case can be recovered as a special case of the MRF. It is however more interesting to choose $\boldsymbol{V}$ in a way which penalizes small variations while not over-penalizing large variations which may be transitions in the image. This point will be discussed in Section 1.4. We do note however, that the choice of $\boldsymbol{V}$ strongly affects the complexity of the problem of minimizing Equation 1.27 as well as the quality of the resulting solution. Once again there is a tradeoff involved.

Having described the two broad classes of estimators, quadratic and non-quadratic, we now turn to examining the algorithms which can be used to obtain the solution of Equation 1.7 for the various choices of J_1 and J_2. We also indicate how to estimate the value of μ.

1.3 Algorithms: Quadratic Criteria

The recent literature contains several algorithms which attempt to solve the image restoration problem for quadratic criteria. In general we find that algorithms which solve the quadratic case are fast and exhibit no problems with convergence

to the minimum as the criterion is convex. However the solution does suffer from smooth transitions. Here we examine algorithms which minimize Equation 1.7 in the quadratic case. We consider in particular conjugate gradient methods and Kalman filtering which have appeared frequently in the recent literature. Another approach is the frequency domain Wiener filter described in (Rathee, Zoly and Overton 1992b). However this will not be developed since here we are interested in spatial domain approaches as ultimately we wish to filter our image adaptively based on local characteristics present in the spatial domain.

1.3.1 Conjugate Gradient Methods

The conjugate gradient has been extensively studied in the context of the least squares problem $\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|$ where $\boldsymbol{H}$ is symmetric. In Equation 1.2, the matrix $\boldsymbol{H}$ is not symmetric, however we note that the least squares solution given by :

$$\hat{\boldsymbol{x}}(\boldsymbol{H}^t\boldsymbol{H})^{-1}\boldsymbol{H}^t\boldsymbol{y} \tag{1.28}$$

is also the solution to the following problem:

$$\hat{\boldsymbol{x}} = \min_{\boldsymbol{x}} \|\boldsymbol{H}^t\boldsymbol{y} - \boldsymbol{H}^t\boldsymbol{H}\boldsymbol{x}\| \tag{1.29}$$

Therefore, we can assume without loss of generality that $\boldsymbol{H}$ is symmetric. The development elucidated here follows (Golub and Van Loan 1989). In what follows we assume that the matrix $\boldsymbol{H}^t\boldsymbol{H}$ is invertible. At the end of this section,we show how a regularized version of the method can solve the problem when this matrix is not invertible. We begin by solving Equation 1.29 which corresponds to the un-regularized problem.

First we observe that minimizing the function:

$$\phi(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^t\boldsymbol{H}\boldsymbol{x} - \boldsymbol{x}^t\boldsymbol{y} \tag{1.30}$$

is also equivalent to solving Equation 1.29 since the minimum occurs at $\boldsymbol{x} = \boldsymbol{H}^{-1}\boldsymbol{y}$. A straightforward way of minimizing Equation 1.30 is to use steepest descent. From a current point $\boldsymbol{x}_c$ the negative gradient gives the direction in which the function decreases most rapidly. This direction is easily computed:

$$-\nabla\phi(\boldsymbol{x}) = \boldsymbol{r}_c = \boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}_c \tag{1.31}$$

where $\boldsymbol{r}_c$ is the residual which, when nonzero, implies the existence of an α such that $\phi(\boldsymbol{x}_c + \alpha\boldsymbol{r}_c) < \phi(\boldsymbol{x}_c)$. By minimizing $\phi(\boldsymbol{x}_c + \alpha\boldsymbol{r}_c)$ with respect to α we immediately obtain $\alpha = \boldsymbol{r}_c^t\boldsymbol{r}_c/\boldsymbol{r}_c^t\boldsymbol{H}\boldsymbol{r}_c$ and the resulting algorithm in Table 1.1 The conjugate

Table 1.1: Conjugate Gradient:basic algorithm

$k = 0$; $\boldsymbol{x}_0 = 0$; $\boldsymbol{r}_0 = 0$
while $\boldsymbol{r}_k > \epsilon$
  $k = k + 1$
  $\alpha_k = \boldsymbol{r}_{k-1}^t\boldsymbol{r}_{k-1}/\boldsymbol{r}_{k-1}^t\boldsymbol{H}\boldsymbol{r}_{k-1}$
  $\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \alpha_k\boldsymbol{r}_{k-1}$
  $\boldsymbol{r}_k = \boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}_k$
end

gradient algorithm performs well when the matrix $\boldsymbol{H}$ is either well conditioned or has only a few distinct eigenvalues. Since in image restoration $\boldsymbol{H}$ is ill-conditioned these conditions are not met, and variants of the conjugate gradient are used. The main idea lies in considering the transformed system $\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{x}}$ where the new matrix $\tilde{\boldsymbol{H}}$ is either well conditioned or has clustered eigenvalues. Defining $\tilde{\boldsymbol{H}} = \boldsymbol{C}^{-1}\boldsymbol{H}\boldsymbol{C}^{-1}$, $\tilde{\boldsymbol{x}}_k = \boldsymbol{C}\boldsymbol{x}_k$

and $\tilde{\boldsymbol{y}} = \boldsymbol{C}^{-1}\boldsymbol{y}$ where $\boldsymbol{C}$ is a symmetric positive definite matrix leads directly to the algorithm in Table 1.2. where the symbol * indicates the complex conjugate transpose

Table 1.2: Conjugate Gradient on Transformed System

$k = 0 : \tilde{\boldsymbol{x}}_0 = 0 : \tilde{\boldsymbol{r}}_0 = \tilde{\boldsymbol{y}}$
while $\boldsymbol{r}_k > \epsilon$
  $k = k + 1$
  if k=1
    $\tilde{\boldsymbol{p}}_1 = \tilde{\boldsymbol{r}}_0$
  else
    $\beta_k = \tilde{\boldsymbol{r}}_{k-1}^t \boldsymbol{r}_{k-1} / \boldsymbol{r}_{k-2}^t \boldsymbol{r}_{k-2}$
    $\tilde{\boldsymbol{p}}_k = \tilde{\boldsymbol{r}}_{k-1} + \beta_k \tilde{\boldsymbol{p}}_{k-1}$
  end
  $\alpha_k = \tilde{\boldsymbol{r}}_{k-1}^t \tilde{\boldsymbol{r}}_{k-1} / \tilde{\boldsymbol{p}}_k^t \boldsymbol{C}^{-1} \boldsymbol{H} \boldsymbol{C}^{-*} \tilde{\boldsymbol{p}}_k$
  $\tilde{\boldsymbol{x}}_k = \tilde{\boldsymbol{x}}_{k-1} + \alpha_k \tilde{\boldsymbol{p}}_k$
  $\tilde{\boldsymbol{r}}_k = \tilde{\boldsymbol{r}}_{k-1} - \alpha_k \boldsymbol{C}^{-1} \boldsymbol{H} \boldsymbol{C}^{-*} \tilde{\boldsymbol{p}}_k$
end
$\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}_k$

of a matrix.

It is possible to remove all references to the matrix $\boldsymbol{C}^{-1}$ by first defining $\tilde{\boldsymbol{p}}_k = \boldsymbol{C}\boldsymbol{p}_k$, $\tilde{\boldsymbol{x}} = \boldsymbol{C}\boldsymbol{x}_k$, then defining the pre-conditioner $\boldsymbol{M} = \boldsymbol{C}\boldsymbol{C}^*$ and letting $\boldsymbol{z}_k$ be the solution to the equation $\boldsymbol{M}\boldsymbol{z}_k = \boldsymbol{r}_k$. This leads to the algorithm for the preconditioned conjugate gradient in Table 1.3. For algorithm 1.3 to be effective, the equation $\boldsymbol{M}\boldsymbol{z} = \boldsymbol{r}$ must be easily solved and in order for the system to converge rapidly, it is necessary that the matrix $\boldsymbol{M}^{-1}\boldsymbol{H} \approx \boldsymbol{I}$. The latter condition is equivalent to having a matrix $\tilde{\boldsymbol{H}}$ with clustered eigenvalues. In the recent literature, various methods of choosing the matrix $\boldsymbol{M}$ have been proposed and algorithms have been developed for the rapid calculation of the solution of 1.29 without explicit calculation of $\boldsymbol{M}$ or the transformed system.

In (Nagy, Plemmons and Torgersen 1996) a method based on approximate inverse

Table 1.3: Preconditioned Conjugate Gradient

$k = 0$; $\boldsymbol{x}_0 = 0$; $\boldsymbol{r}_0 = \boldsymbol{y}$
while $\boldsymbol{r}_k > \epsilon$
  Solve $\boldsymbol{M}\boldsymbol{z}_k = \boldsymbol{r}_k$
  $k = k + 1$
  if k=1
    $\boldsymbol{p}_1 = \boldsymbol{z}_0$
  else
    $\beta_k = \boldsymbol{r}_{k-1}^t \boldsymbol{z}_{k-1} / \boldsymbol{r}_{k-2}^t \boldsymbol{z}_{k-2}$
    $\boldsymbol{p}_k = \boldsymbol{z}_{k-1} + \beta_k \boldsymbol{p}_{k-1}$
  end
  $\alpha_k = \boldsymbol{r}_{k-1}^t \boldsymbol{z}_{k-1} / \boldsymbol{p}_k^t \boldsymbol{H} \boldsymbol{p}_k$
  $\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \alpha_k \boldsymbol{p}_k$
  $\boldsymbol{r}_k = \boldsymbol{r}_{k-1} - \alpha_k \boldsymbol{H} \boldsymbol{p}_k$
end
$\boldsymbol{x} = \boldsymbol{x}_k$

conditioning was proposed. The idea is to exploit the possibility of using the Fast Fourrier Transform (FFT) to reduce the cost when multiplying by circulant matrices. First, the matrix $\boldsymbol{H}$ in 1.2 must be extended from a Toeplitz block Toeplitz matrix to a block circulant matrix where each block is itself circulant. The new matrix $\boldsymbol{H}$ takes the form:

$$\boldsymbol{H} = \left[\begin{array}{ccccc|ccc} \boldsymbol{H}_0 & 0 & \cdots & \cdots & \cdots & \boldsymbol{H}_\gamma & \cdots & \boldsymbol{H}_1 \\ \boldsymbol{H}_1 & \boldsymbol{H}_0 & 0 & \cdots & \cdots & & \ddots & \vdots \\ \vdots & & \ddots & & \cdots & & & \boldsymbol{H}_\gamma \\ \boldsymbol{H}_\gamma & & & \ddots & & & & \\ & \ddots & & & \boldsymbol{H}_0 & & & \\ & & \ddots & & \boldsymbol{H}_1 & \boldsymbol{H}_0 & & \\ & & & \ddots & \vdots & \vdots & \ddots & \\ & & & & \boldsymbol{H}_\gamma & \boldsymbol{H}_{\gamma-1} & \cdots & \boldsymbol{H}_0 \end{array}\right] \tag{1.32}$$

We can write:

$$\boldsymbol{BP} = \left[\boldsymbol{H} \mid \hat{\boldsymbol{H}}\right] \Leftrightarrow \boldsymbol{B} = \left[\boldsymbol{H} \mid \hat{\boldsymbol{H}}\right] \boldsymbol{P}^* \tag{1.33}$$

and

$$\boldsymbol{P}^* \boldsymbol{B}^{-1} = \begin{bmatrix} \boldsymbol{A}^* \\ \hat{\boldsymbol{A}}^* \end{bmatrix} \Rightarrow \boldsymbol{B}^{-1} = \boldsymbol{P} \begin{bmatrix} \boldsymbol{A}^* \\ \hat{\boldsymbol{A}}^* \end{bmatrix} \tag{1.34}$$

where the matrix $\boldsymbol{P}$ is a permutation matrix $\boldsymbol{H}$ is the basic Toeplitz block Toeplitz matrix from 1.2 ,$\hat{\boldsymbol{H}}$ contains the columns that are added on to $\boldsymbol{H}$, and $\boldsymbol{B}$ is the unpermuted block circulant matrix. We note that the permutation matrix $\boldsymbol{P}$ is unitary. That is $\boldsymbol{P}^{-1} = \boldsymbol{P}^*$.

The aim is to construct an approximate inverse preconditioner $\boldsymbol{K} = \boldsymbol{M}^{-1}$ such that $\boldsymbol{K}\boldsymbol{H}^t\boldsymbol{H} \approx \boldsymbol{I}$. It is shown in (Nagy et al. 1996) that $\boldsymbol{K} = \boldsymbol{A}^*\boldsymbol{A}$ is a suitable choice and it is demonstrated that $(\boldsymbol{A}^*\boldsymbol{A})(\boldsymbol{H}^t\boldsymbol{H}) = \boldsymbol{I} + \boldsymbol{R}$ where $rank(\boldsymbol{R}) \leq n\gamma + n\beta - \beta\gamma$ where n^2 is the number of pixels in the image; we recall that γ and β are the dimensions of the PSF. However this is only and upper bound and in practice it is frequently found that the actual value of the rank is much lower due to the clustering of the eigenvalues of the PSF. The value of the rank of the matrix $\boldsymbol{R}$ is significant since it is roughly equal to the number of iterations that will be required for convergence.

It must be noted that the computation of solution given by Table 1.3 is dominated by the matrix vector products $\boldsymbol{M}^{-1}\boldsymbol{r}$ and $\boldsymbol{Hp}$. One motivation for the choice of the above preconditioner is the fact that these computations can be carried out rapidly by means of a 2-D FFT algorithm. In this particular development the products $\boldsymbol{M}^{-1}\boldsymbol{r}$ and $\boldsymbol{Hp}$ are replaced by $\boldsymbol{Kr}$ and $\boldsymbol{H}^*\boldsymbol{Hp}$.

The method presented up to this point yields a solution to the least squares

problem 1.29. However solutions to this problem are unstable and must be regularized in order to produce acceptable results. It turns out that the above algorithm can be easily adapted to the regularized problem. As explained at the beginning of this literature survey, minimizing Equation 1.7 with J_1 and J_2 chosen to be quadratic is equivalent to solving:

$$\min \left\| \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix} - \begin{bmatrix} \boldsymbol{H} \\ \nu \boldsymbol{L} \end{bmatrix} x \right\|_2 \tag{1.35}$$

with $\boldsymbol{L} = \boldsymbol{Q}^t\boldsymbol{Q}$ and $\nu = \sqrt{\mu}$. Essentially the only modification resulting from introducing regularization into the problem is that the matrix $\boldsymbol{B}$ defined in Equation 1.33 changes so that we have:

$$\boldsymbol{B}^*\boldsymbol{B} = \boldsymbol{B}_{\boldsymbol{H}}^*\boldsymbol{B}_{\boldsymbol{H}} + \mu^2\boldsymbol{I} = \boldsymbol{F}^*(|\Lambda_{\boldsymbol{H}}|^2 + \mu^2\boldsymbol{I})\boldsymbol{F} \tag{1.36}$$

where $\boldsymbol{B}_{\boldsymbol{H}}$ is the matrix $\boldsymbol{B}$ defined in Equation 1.33 associated with the circulant extension of $\boldsymbol{H}$. the operator $\boldsymbol{L}$ has been taken to be the identity matrix, and the right hand side is the standard factorization of a block circulant matrix in terms of its eigenvalues ($|\Lambda_{\boldsymbol{H}}|$) and Fourrier matrices as in (Andrews and Hunt 1977). These eigenvalues can be computed in roughly $n \log n$ operations by using a Fast Fourrier Transform (FFT) algorithm. Exploiting the Fourrier transforms where applicable and using the approximate inverse preconditioner $\boldsymbol{M}^{-1} = \boldsymbol{B}^*\boldsymbol{B}$ leads to the algorithm in Table 1.4. In this table fft and ifft are used to denote the Fast Fourrier transform and its inverse respectively. x^c denotes the component-wise complex conjugate of a vector, $\circ$ denotes component-wise multiplication and $\varnothing$ is used for component-wise division. Finally, the *rotate* function is defined as follows such that $b = rotate(a, k)$

where a is an $n \times n$ matrix gives:

$$b_{i,j} = a_{((i+k-1) mod\ n)+1,((j+k-1) mod\ n)+1} \tag{1.37}$$

In words, the elements of the matrix b are obtained by rotating the columns of a to the right by k columns and then rotating the rows downward by k columns. When applied in practice, the algorithm in Table 1.4 converges in 4 or 5 iterations as compared to the roughly 50 iterations required for the conjugate gradient without pre-conditioning. The algorithm is effective and performs comparably to the algorithms presented in the main section of this thesis since the same criteria is being minimized. However conjugate gradient methods are not effective in processing images in 3D since the matrices are too large and the computational load too heavy. Furthermore, the type of conjugate algorithms presented here treat the image globally and consequently cannot filter the image based on local characteristics such as the presence or absence of a transition. An adaptive filter which is able to perform such filtering is presented in Section 3.

Table 1.4: Regularized Preconditioned Conjugate Gradient

$\boldsymbol{x}$ is an $m \times m$ matrix (image to be estimated)
$\boldsymbol{y}$ is an $n \times n$ matrix (observed image)
$\boldsymbol{h}$ is an $n \times n$ matrix (point spread function)
$\boldsymbol{y} = \boldsymbol{h} * \boldsymbol{x}$
$\boldsymbol{x}_e = \begin{bmatrix} \boldsymbol{x} & 0 \\ 0 & 0 \end{bmatrix}$, x_e is an $n \times n$ matrix
$\boldsymbol{h}_r = rotate(\boldsymbol{h}, n/2)$
$\hat{\boldsymbol{y}} = \text{fft2}(\boldsymbol{y}), \hat{\boldsymbol{g}} = \hat{\boldsymbol{h}}^c \circ \hat{\boldsymbol{h}} + \mu^2$
$\hat{\boldsymbol{r}}_e = \hat{\boldsymbol{h}}^c \circ \hat{\boldsymbol{y}} - \hat{\boldsymbol{g}} \circ \hat{\boldsymbol{f}}_e$
$\boldsymbol{r}_e = \text{ifft2}(\hat{r}_e), \boldsymbol{r} = \boldsymbol{r}_e(1:m, 1:m)$
$k = 0$
while $\|r\| > \epsilon$
$\boldsymbol{r}_e = \begin{bmatrix} r & 0 \\ 0 & 0 \end{bmatrix}$, $\hat{r}_e = \text{fft2}(\boldsymbol{r}_e)$
$\hat{\boldsymbol{z}}_e = \hat{r}_e \varnothing \hat{g}$
$\boldsymbol{z}_e = \text{ifft2}(\hat{\boldsymbol{z}}_e), \boldsymbol{z} = \boldsymbol{z}_e(1:m, 1:m)$
$\rho = \text{sum2}(\boldsymbol{r} \circ \boldsymbol{w})$
$k = k + 1$
if $k = 1$
    $\boldsymbol{p} = \boldsymbol{z}$
else
    $\beta = \rho / \rho_{old}$
    $\boldsymbol{p} = \boldsymbol{z} + \beta \boldsymbol{p}$
end
$\boldsymbol{p}_e = \begin{bmatrix} \boldsymbol{p} & 0 \\ 0 & 0 \end{bmatrix}$, $\hat{\boldsymbol{p}} = \text{fft2}(\boldsymbol{p}_e)$
$\hat{\boldsymbol{w}}_e = \hat{\boldsymbol{g}} \circ \hat{\boldsymbol{p}}_e$
$\boldsymbol{w}_e = \text{ifft2}(\hat{\boldsymbol{w}}_e), \boldsymbol{w} = \boldsymbol{w}_e(1:m, 1:m)$
$\alpha = \rho / sum2(p \circ w)$
$\boldsymbol{x} = \boldsymbol{x} + \alpha \boldsymbol{p}$
$\rho_{old} = \rho$
$\boldsymbol{r} = \boldsymbol{r} - \alpha \boldsymbol{w}$
end

1.3.2 Kalman Filtering

In the main section of this thesis we show one way in which Kalman filtering can be used to minimize the quadratic criterion. In recent years, other methods have been proposed, also based on the Kalman filter. One particularly interesting approach is the reduced update Kalman filter (RUKF) proposed first by Woods in (Woods and Radewan 1977) for the general 2D filtering problem and then adapted to the problem of image restoration by Woods in (Woods and Ingle 1981). The algorithm uses an auto-regressive model for the image which, as we will see, is equivalent to a Gaussian density $p(\boldsymbol{x})$. We recall that introducing a Gaussian density in the stochastic framework for $p(\boldsymbol{x})$ is equivalent to introducing a quadratic for J_2 in the deterministic framework.

We begin by presenting the problem in one dimension. First we state the auto-regressive model for our vector to be estimated:

$$x(m) + \sum_{1}^{p} f(k)x(m-k) = u(m) \tag{1.38}$$

where $u(m)$ is a Gaussian random variable and f is a scalar while p is the order of the process. Such a model can also be thought of as a filtering process in which the signal $x(n)$ is filtered by an auto-regressive filter to yield a Gaussian process $u(n)$. We will now demonstrate how to calculate the covariance matrix of $\boldsymbol{x}$ from the coefficients of f_k and also how to find f_k from the covariance matrix of $\boldsymbol{x}$. We follow the derivation given by Proakis (Proakis and Manolakis 1988). We begin by multiplying by $x(n-m)$

and taking the expected value.

$$E[x(n)x(n-m)] = -\sum_{k=1}^{p} f_k E[x(n-k)x(n-m)] + E[u(n)x(n-m)] \quad (1.39)$$

$$\gamma_{xx}(m) = -\sum_{k=1}^{p} f_k \gamma_{xx}(m-k) + \gamma_{ux}(m) \quad (1.40)$$

where γ_{xx} and γ_{ux} are auto-correlation and cross-correlation functions respectively. We develop γ_{ux}:

$$\gamma_{ux} = E[x(n)u(n+m)] \quad (1.41)$$

$$= E[\sum_{k=0}^{\infty} h(k)w(n-k)w(n+m)] \quad (1.42)$$

$$= \sigma_u^2 h(-m) \quad (1.43)$$

where $h(m)$ is 0 for $m < 0$ by causality. It now follows that:

$$\gamma_{xx}(m) = \begin{cases} -\sum_{k=1}^{p} f_k \gamma_{xx}(m-k) & m > 0 \\ -\sum_{k=1}^{p} f_k \gamma_{xx}(m-k) + \sigma_u^2 & m = 0 \\ \gamma_{xx}(-m) & m < 0 \end{cases} \quad (1.44)$$

We can thus write in matrix form:

$$\begin{bmatrix} \gamma_{xx}(0) & \gamma_{xx}(1) & \gamma_{xx}(2) & \cdots & \gamma_{xx}(p) \\ \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma_{xx}(1) & \cdots & \gamma_{xx}(p-1) \\ \vdots & \vdots & \vdots & & \vdots \\ \gamma_{xx}(p) & \gamma_{xx}(p-1) & \gamma_{xx}(p-2) & \cdots & \gamma_{xx}(0) \end{bmatrix} \begin{bmatrix} 1 \\ f_1 \\ \vdots \\ f_p \end{bmatrix} = \begin{bmatrix} \sigma_u^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (1.45)$$

The above matrix equation contains Yule-Walker equations which relate the elements

of the covariance matrix γ_{xx} to the auto-regressive model given in 1.38. With these equations we see that an auto-regressive model with Gaussian noise u_n is equivalent to specifying an *a priori* $p(\boldsymbol{x})$ which is Gaussian with the elements of the covariance matrix chosen to solve the Yule Walker equations. These equations are solved by the Levinson-Durbin algorithm (Levinson 1947, Durbin 1959). Given the filter coefficients we can solve for the elements of the covariance matrix by using the inverse Levinson algorithm. Consequently the auto-regressive process is a parameterization of the covariance matrix.

This equivalence exists only when the Paley-Wiener theorem is satisfied. That is we require:

$$\int_{-\pi}^{\pi} |\log |H(\omega)|| d\omega < \infty \tag{1.46}$$

Where $H(\omega)$ is the spectrum of the filter. This is equivalent to requiring that the spectrum be non-zero everywhere except at a finite number of points.

Having demonstrated the equivalence between an auto-regressive model and the specification of a covariance matrix of a Gaussian process, we are now in position to develop the Kalman filtering approach. We first note that Equation 1.38 can be put into matrix form as given in Equations 1.47 and 1.49. The general system model

which is well-known in the stochastic control domain is now given in matrix form by:

$$\boldsymbol{x}(m) = \boldsymbol{F}\boldsymbol{x}(m-1) + \boldsymbol{g}u(m) \tag{1.47}$$

$$\boldsymbol{y}(m) = \boldsymbol{H}\boldsymbol{x}(m) + \boldsymbol{v}(m) \tag{1.48}$$

$$\boldsymbol{F} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & 1 \\ f_1 & f_2 & \dots & \dots & \dots & f_p \end{bmatrix} \tag{1.49}$$

$$\boldsymbol{x}(m) = \begin{bmatrix} x_{m-p} \\ x_{m-p+1} \\ \vdots \\ x_m \end{bmatrix} ; \boldsymbol{g} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{1.50}$$

where $\boldsymbol{x}$ is the state of the system, $\boldsymbol{y}$ is the observed vector, $\boldsymbol{H}$ and $\boldsymbol{F}$ are known matrices, and $u(m)$ and $\boldsymbol{v}$ are sequences of Gaussian white noise independent of each other and independent of $\boldsymbol{x}_k$ such that we have:

$$E[u(m)] = E[\boldsymbol{v}(m)] = 0 \tag{1.51}$$

$$E[u(m)u(k)] = q_u\delta_{mk} \tag{1.52}$$

$$E[\boldsymbol{v}(m)\boldsymbol{v}(k)^t] = \boldsymbol{Q}_v\delta_{mk} \tag{1.53}$$

$$E[\boldsymbol{v}(m)u(k)^t] = 0 \tag{1.54}$$

where $\boldsymbol{Q}_u$ and $\boldsymbol{Q}_v$ are covariance matrices. Equation 1.47 is our auto-regressive model for the image while Equation 1.48 is the observation equation.

This is the same set of equations as those considered in Section 2.1 with the

notation slightly changed so as to be easily extended to two dimensions. We seek a solution which maximizes $E[\boldsymbol{x}|y(1), y(2), ..., y(k)]$ where $y(k)$ is the current pixel. That is, we wish to maximize the probability of $\boldsymbol{x}$ given all our observations up to and including the current point. In the case of Gaussian noise, which we assume here, this is equivalent to the MAP solution given by:

$$x_{MAP} = \arg\max_{x} p(x|y) = \arg\max_{x} p(y|x)p(x) \tag{1.55}$$

where $p(\boldsymbol{y}|\boldsymbol{x})$ and $p(\boldsymbol{x})$ are given by 1.16 and 1.17 respectively. Since these are Gaussians, we immediately observe that we are equivalently solving Equation 1.7 with J_1 and J_2 chosen as quadratics. Computing this solution can be done by using a Kalman smoother. However, this requires about 4 times as many computations as the Kalman filter which we develop in Section 2.1. The problem with the Kalman filter, is that it is sub-optimal in the sense that $\hat{\boldsymbol{x}}_k$ is estimated from $y_1 \ldots y_k$ and not $y_1 \ldots y_N$ where N is the number of observations. It is clear that the points updated at the end have the benefit of all the observations while those calculated at the beginning have very few observations. In practice this sub-optimality is not significant as a delay can be artificially be introduced so that the current estimate is made with the benefit of future observations. The solution for this problem is developed in detail in Section

2.1. We quote the solution here:

$$\hat{\boldsymbol{x}}_b(m) = \boldsymbol{F}\hat{\boldsymbol{x}}_a(m-1) \tag{1.56}$$

$$\boldsymbol{P}_b(m) = \boldsymbol{F}\boldsymbol{P}_a(m-1)\boldsymbol{F}^t + q_u\boldsymbol{g}\boldsymbol{g}^t \tag{1.57}$$

$$\boldsymbol{K}(m) = \boldsymbol{P}_b(m)\boldsymbol{H}^t(\boldsymbol{H}\boldsymbol{P}_b(m)\boldsymbol{H}^t + \boldsymbol{Q}_v)^{-1} \tag{1.58}$$

$$\hat{\boldsymbol{x}}_a(m) = \hat{\boldsymbol{x}}_b(m) + \boldsymbol{K}(m)[\boldsymbol{y}(m) - \boldsymbol{H}\hat{\boldsymbol{x}}_b(m)] \tag{1.59}$$

$$\boldsymbol{P}_a(m) = [\boldsymbol{I} - \boldsymbol{K}(m)\boldsymbol{H}]\boldsymbol{P}_b(m) \tag{1.60}$$

where

$$\boldsymbol{P}_i(m) = \boldsymbol{E}[(\boldsymbol{x} - \hat{\boldsymbol{x}})_i(\boldsymbol{x} - \hat{\boldsymbol{x}})_i^t], i = a, b \tag{1.61}$$

where the subscripts a and b indicate "after" and "before" updating respectively. We now examine the extension to 2D and point out the additional sub-optimalities that are introduced.

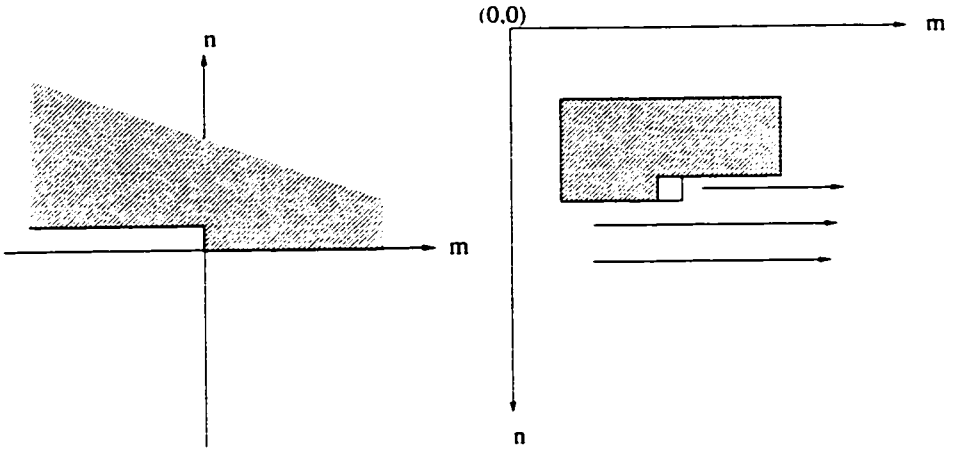
In extending the 1D auto-regressive model to two dimensions, we are faced with the problem of expressing the current pixel in terms of "past" pixels. Therefore an ordering of pixels must be defined. This is usually taken to be the raster scan ordering in which pixels are visited in one line from left to right and then the line below is visited starting from the left once again. This ordering immediately introduces asymmetry into the system since results will be different if we first rotate the image.

Once the pixels have been ordered, a causal auto-regressive model, compatible with the recursive Kalman filter, must be introduced. Unlike in 1D, we no longer have the equivalence between an auto-regressive model and a covariance matrix since in 2D auto-regressive models are non-causal. However, approximations can be found using spectral factorization techniques. One of them is the non-symmetric half plane

(NSHP) recursive model given by:

$$x(m,n) = \sum_{\boldsymbol{R}_\oplus} c_{kl} x(m-k, n-l) + w(m,n) \tag{1.62}$$

where $w(m,n)$ is a white noise Gaussian noise field, $\boldsymbol{R}_\oplus = k \geq 0, l \geq 0 \cup \{k < 0, l > 0\}$ and $c_{00} = 0$. This NSHP model is the result of work by Woods (Woods and Ekstrom 1975) and Ekstrom (Ekstrom M. P. 1976). Since we use a raster scan ordering of pixels, the NSHP model allows us to transform a 2D model into a 1D causal auto-regressive model since with the NSHP model the present point is written in terms of the past. We note that in (Woods and Ekstrom 1975) it is shown that under weak conditions a 2D spectrum can be factored into NSHP factors. The support of such a NSHP filter is shown in figure 3(b) along with a diagram of the associated recursive filtering. The support of the filter is given by $\{-M \geq k \geq 0, N \geq l \geq 0\} \cup \{-M \leq k < 0, N \geq l > 0\}$ where N and M are determined by the spectral factorization. This factorization induces a sub-optimality in that the NSHP filter is equivalent to the spectrum only when the dimensions N and M are infinite. By truncating, we immediately introduce sub-optimality. It is however shown in (Ekstrom M. P. 1976) that the filter can be designed to be stable and as accurately as desired if we take the support (M, N) to be sufficiently large. However designing such filters is not an easy task and furthermore a tradeoff is involved between the computational complexity and the accuracy of the filter. A further sub-optimality is introduced in that the model no longer holds at the edges of the image. The model must be adapted at the boundaries. One possible solution is to add zeros around the edges of the image, but this introduces new observations into the system and consequently yields an additional sub-optimality.



(a) NSHP support region (b) Recursion of filter

Figure 1.3: NSHP support region and Recursion of filter

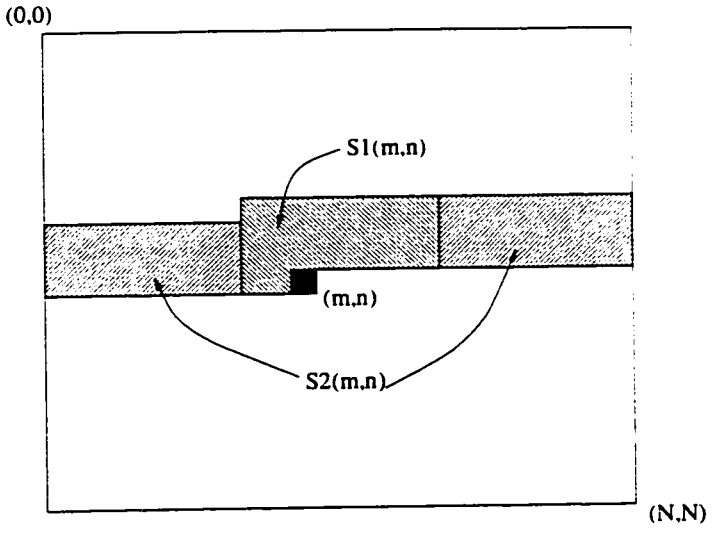


Figure 1.4: Definition of the regions S1 and S2

With the Kalman filter, the size of the matrix $\boldsymbol{P}$ is related to the size of the state being estimated. This state is related directly to the size of the NSHP filter being used. The complete state is composed of both the regions S1 and S2 illustrated in Figure 1.4. Clearly the complete state includes several lines of the image and thus contains a large number of points to be updated. In order to save on calculations, the RUKF was developed in which only the small section marked S1 in Figure 1.4 is updated. The state vector can be written as:

$$\boldsymbol{x}(m,n) \triangleq \left[\frac{\boldsymbol{x}_1(m,n)}{\boldsymbol{x}_2(m,n)}\right] \tag{1.63}$$

where

$$\begin{aligned}\boldsymbol{x}_1 =&[x(m,n), x(m-1,n), \ldots, x(m-M+1,n);\\ &x(m+M+1,n-1), \ldots, x(m-M+1,n-1);\\ &\ldots : x(m+M+1,n-M), \ldots, x(m-M+1,n-M)]^t\end{aligned} \tag{1.64}$$

and $\boldsymbol{x}_2$ contains the rest of the state. A sub-optimality is introduced by updating the small section $\boldsymbol{x}_1$ rather than the entire vector which is updated with the standard filter. However, we expect that the effect of this sub-optimality is small since points near the observation are all updated, and it is only points further away which are neglected.

Using the NSHP model, the new system takes the following form with a scalar observation in two dimensions.

$$\boldsymbol{x}(m,n) = \boldsymbol{C}\boldsymbol{x}(m-1,n) + \boldsymbol{w}(m,n) \tag{1.65}$$

$$y(m,n) = \boldsymbol{h}^t\boldsymbol{x}(m,n) + v(m,n) \tag{1.66}$$

where $\boldsymbol{h}_1^t = [h_{0,0}, \ldots, h_{-M,0}; h_{M,-1}; \ldots : h_{M,-M}, \ldots, h_{-M,-M}]$ and $\boldsymbol{h}_2^t = 0$. $\boldsymbol{h}$ is then the matrix composed of these two partitions such that the partitions 1 and 2 multiply the corresponding partitions of $\boldsymbol{x}$. Here $\boldsymbol{h}$ is an FIR filter of size $M \times M$ which is made up of samples of the PSF.

In (Woods and Radewan 1977) the scalar Kalman filter which solves the above system is given with a detailed derivation. The equations resemble those of the standard Kalman filter with some important differences with the indexing which stem from the fact that only parts of the state vector are being updated.

We have seen in this section that the RUKF yields results which are sub-optimal in several ways with respect to the optimal calculation of the MAP estimate. However the sub-optimalities are shown in practice to be small. The reason for this is that when a given point is observed, it only gives us information about points in the local neighborhood, usually slightly bigger than the size of the PSF. This is the case since the correlation matrices are usually chosen to model correlation at small distances. This is reasonable since in most real images we expect that nearby points will be similar while points far apart will be un-correlated with one another. Consequently by reducing the points being updated to a local neighborhood slightly bigger than the PSF we expect that little will be lost. Furthermore, though sub-optimalities are introduced by the NSHP, these can be reduced by extending the support of the NSHP filter. However a compromise must be made since enlarging the support increases the volume of calculations. Filters of the same class and presenting similar numerical characteristics while avoiding the sub-optimalities can be developed and are the subject of Section 2.1.

1.4 Algorithms: Non-Quadratic Criteria

We now tun our attention to the non-quadratic case and in particular the subject of Markov random fields. In the last 15 years considerable work has been done in the area of Markov Random Fields (MRFs). The use of MRFs allows images to be explicitly modeled as homogeneous zones separated by abrupt discontinuities. This avoids the excessive smoothing of edges associated with quadratic criteria in which the object to be restored is assumed to be Gaussian. We recall that the equation being minimized is the following

$$J(x) = \frac{\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2}{2\sigma_n^2} + \lambda \sum_{c \in C} \boldsymbol{V}_c(\boldsymbol{x}) \tag{1.67}$$

which is Equation 1.27 restated. We are interested in various choices of the second term. We recall that by choosing the second term to be quadratic we recover the quadratic case. Here we will be interested in other choices which will allow us to explicitly model transitions in the image. We now consider choices of $\boldsymbol{V}_c$ in the following historical review of developments in the of MRFs. We will find that once again, the choice of $\boldsymbol{V}_c(x)$ will strongly influence the quality of the solution as well as the complexity of the algorithm and computational cost involved in finding the solution.

The landmark work of Donald and Stuart Geman in 1984 (Geman and Geman 1984) yielded several important innovations in image restoration using MRFs. They proposed modeling the image as an MRF with a hidden line field. Thus the image is represented by $\boldsymbol{X}(x, l)$ where both the pixel intensities and the line field are estimated. In this model, the pixel intensities are chosen to be discrete valued, and the line field is a "hidden" binary quantity indicating the presence or absence of an edge. These lines

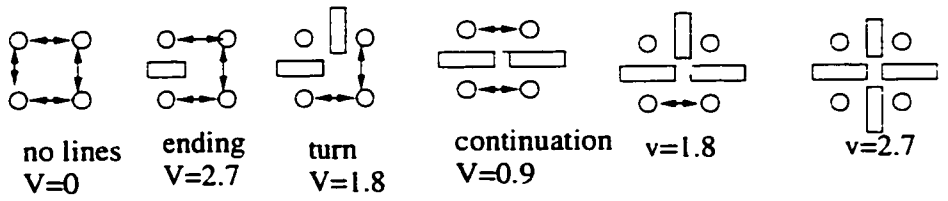


Figure 1.5: Line field examples

are placed between neighboring pixel sites and a penalty function is then associated with various configurations as illustrated in Figure 1.5. If an edge exists their is no interaction between the associated pixels. Since in an image we expect contours to form continuous paths, lines which end nowhere are penalized more. The case with no lines which corresponds to a homogeneous zone is penalized the least while continuous lines have intermediate penalties.

Methods based on the conjugate gradient do not exist to minimize Equation 1.27 in the case of discrete intensities for x with a coupled line field. Consequently, Geman and Geman in 1984 proposed a "simulated annealing" algorithm in which estimates are obtained by sampling a local probability distribution. Choosing these local probability distributions to be consistent with the Gibb's distribution is not straightforward. More precisely, it is required that the local probability distribution of the intensity given the line field $p(x|l)$ be consistent with the global specification $V_c(x, l)$. However, the Hammersley-Clifford (Geman and Geman 1984) theorem demonstrates that it is possible to choose such distributions to be consistent with the global Gibb's potential. The local distributions depend on a parameter T. For a given value of T, the Gibb's sampler is used to obtain an image from the local densities $p(\boldsymbol{x}|\boldsymbol{y})$. At high values of T, the probability distributions are essentially uniform. As T is lowered, the distributions $p(\boldsymbol{x}|\boldsymbol{y})$ favor the maxima of $p(\boldsymbol{x}|\boldsymbol{y})$ more and more.

The simulated annealing algorithm generates a sequence of images for successively decreasing values of T. It has been proven that if the schedule for decreasing T is

chosen appropriately, convergence to the MAP estimate results. However convergence is slow and can only be attained for an infinite time frame.

Since the simulated annealing algorithm is computationally expensive and slow in convergence, other algorithms have appeared. Jeng and Woods in 1991 (Jeng and Woods 1991) proposed an image model which was a Gaussian random field with a hidden line field. Unlike Geman's method this allows the modeling of continuous range of pixel intensities. In order to avoid the computational problems associated with simulated annealing, another approach is proposed in which local probabilities are maximized rather than sampled as the temperature is lowered. This leads to more rapid convergence, however this algorithm only converges to local maxima.

Some recent work has been done with functions without a line field. One choice of $V_c(x)$ which has been popular in recent years and has led to interesting results is:

$$V_c(x) = \phi(u_c) \tag{1.68}$$

$$u_c = \frac{x_i - x_j}{d(i,j)} \tag{1.69}$$

where $d(i,j)$ is the distance between neighboring pixels as illustrated in Figure 1.2. The set of neighbors $\mathcal{N}$ is often chosen to be the set of immediately adjacent pixels including diagonally adjacent pixels also illustrated in figure 1.2. Three types of functions have been studied extensively.

1. Gaussians which yield the same solutions as the Kalman filter discussed later in this thesis. This is the case since using a $\phi(u_c)$ which is quadratic yields the same Equation 1.7 to be minimized if we choose the matrix $\boldsymbol{Q}$ in Equation 1.13 appropriately.

2. The hard-limited quadratic proposed by Blake and Zisserman (Blake and Zis-

serman 1987), and the function $\phi(u) = u^2/(1+u^2)$ proposed by Geman and McClure (Geman and McClure 1987).

3. Functions which are quadratic near the origin, but are asymptotically linear such as the Huber function (Stevenson and Delp 1990) and Green's log cosh function (Green 1990).

Some of these functions are illustrated in Figure 1.6. Using a quadratic function for $\phi(u_c)$ has the disadvantage of over-penalizing large differences which is undesirable as these may represent transitions in the image. On the other hand, Blake and Zisserman's hard-limited quadratic is a non-convex function and minimizing the resulting criteria is difficult since local minima may exist. In order to perform the minimization, the graduated non-convexity (GNC) algorithm has been proposed. In this algorithm a convex function is initially chosen and minimized. Then this function is slowly transformed into the non-convex function. The solution is computed iteratively and at each iteration, the minimum is sought and then used as the starting point for the search on the next function. Choosing a suitable sequence of functions is critical for fast convergence. This problem is discussed in (Blake and Zisserman 1987).

The third type of function is an interesting compromise since it retains the convexity property while not over-penalizing large transitions. Since convex methods present no problems with local minima, the conjugate gradient methods presented in the previous section can be easily adapted to find the minimum. However these methods tend to be computationally expensive since at each iteration the entire image is being updated. Furthermore they require a large amount of memory. Consequently, attention has turned to a class of algorithms known as single site update algorithms (SSUAs) (Sauer and Bouman 1993). In such algorithms pixels are visited in turn and updated based on local information. The resulting algorithms are highly paral-

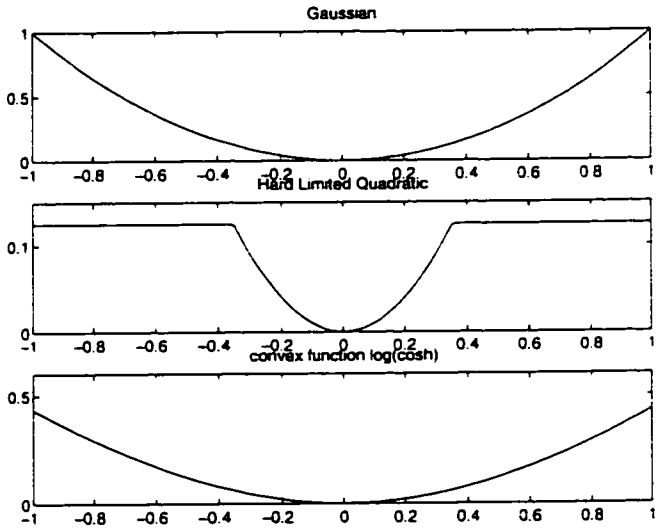


Figure 1.6: Different Types of Energy Functions

lel. These algorithms have been used for the convex and non-convex cases. In the convex case they converge to the minimum. In the non-convex case problems with local minima can be encountered. We now consider one recent approach applicable only in the common case of a separable PSF.

In 1996 Brette and Idier proposed an optimized SSUA for image deblurring (Brette and Idier 1996). Following their development we present the algorithm first for the case where $\phi(u_c) = u^2$ that is the Gaussian case, and then show how to extend to general type convex functions which penalize large transitions less. In the Gaussian case, the minimum of 1.27 for a pixel x_{ij} is reached simply by taking the derivative and setting to zero. This yields for the minimum m_{ij}:

$$m_{ij} = x_{ij} + \frac{[\boldsymbol{H}^t\boldsymbol{y}]_{ij} - [\boldsymbol{H}^t\boldsymbol{H}\boldsymbol{x}]_{ij} + 2\sigma^2 \sum_{r\in\boldsymbol{N}} \lambda_{r,ij}(x_r - x_{ij})}{[\boldsymbol{H}^t\boldsymbol{H}_{ij,ij}] + 2\sigma^2 \sum_{r\in\boldsymbol{N}} \lambda_{r,ij}} \tag{1.70}$$

where N is used to denote the neighbors of the current pixel. Computations can be saved by pre-calculating $[\boldsymbol{H}^t\boldsymbol{y}]_{ij}$ and stored it since it does not depend on the current x_{ij}. Also as $\boldsymbol{H}$ is assumed separable, it can be written as $\boldsymbol{H} = \boldsymbol{v}^t\boldsymbol{h}$ which

allows an efficient computation of $[\boldsymbol{H}^t\boldsymbol{H}\boldsymbol{x}]_{ij}$ if we first assume that $(\boldsymbol{g},\boldsymbol{w})$ are the autocorrelation functions of the horizontal and vertical components respectively, and then define $\boldsymbol{R}_k^{ij}$ as follows:

$$\boldsymbol{R}_k^{ij} = \sum_{l=-(q-1)}^{q-1} g_l x_{i+k,j+l} \tag{1.71}$$

we obtain

$$(\boldsymbol{H}^t\boldsymbol{H}\boldsymbol{x})_{ij} = \sum_{k=-(q-1)}^{q-1} w_k \boldsymbol{R}_k^{ij} \tag{1.72}$$

By exploiting the symmetry in the vectors $(\boldsymbol{g},\boldsymbol{w})$ and also using the fact that $\boldsymbol{R}_k^{ij}$ is shift invariant with $\boldsymbol{R}_k^{i+1,j} = \boldsymbol{R}_{k+1}^{i,j}$ we obtain the algorithm in Table 1.5. The parameter w is a relaxation coefficient which improves the rate of convergence (Press, Teukolsky, Vetterling and Flannery 1992). Algorithm 1.5 applies to quadratic forms, however we can adapt the criteria $\boldsymbol{J}(\boldsymbol{x})$ by adding an auxiliary variable as in (Geman and Reynolds 1992) and then use the duality principle developed by Luenberger in (Luenberger 1969) so that the same method can be used for general convex functions $\phi(u_c)$. This is accomplished by first redefining

$$\phi(u) = \inf_{l\in\boldsymbol{R}^+} lu^2 + \psi(l) \tag{1.73}$$

With this definition it follows that

$$\boldsymbol{J}(\boldsymbol{x},\boldsymbol{l}) = \frac{\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|^2}{2\sigma_n^2} + \lambda \sum_{r,s\in C} l_{rs}(x_r - x_s)^2 + \psi(l_{rs}) \tag{1.74}$$

has the same minimum as the $\boldsymbol{J}(x)$ defined in 1.27. The minimum can be obtained by alternating the minimization with respect to x and l. Since as a function of $\boldsymbol{x}$, $\boldsymbol{J}(x,l)$ is quadratic, the algorithm in Table 1.5 can be used. As a function of l, $\boldsymbol{J}(x,l)$

Table 1.5: Single Site Update Algorithm

1. Initialize: calculate and store $\boldsymbol{H^t y}$, $\boldsymbol{g}$ and $\boldsymbol{w}$.
2. For each column $j = 1, \ldots, \boldsymbol{N}$ initialize $\boldsymbol{R^{0j}}$
   (a) For each pixel $i = 1, \ldots, \boldsymbol{M}$: in the jth column
   (b) Apply the mask $[\boldsymbol{H^t H x}]_{ij} = \boldsymbol{w^t R^{ij}}$
   (c) Add Gibbs and retro-filtered contributions to compute m_{ij}
   (d) Update x_{ij} subject to domain constraints $x_{ij}^{new} \leftarrow x_{ij} + w(m_{ij} - xij)$
   (e) Update $\boldsymbol{R_0^{ij}} \leftarrow \boldsymbol{R_0^{ij}} + w_0(x_{ij}^{new} - x_{ij})$ to take x_{ij}^{new} into account.
   (f) $\boldsymbol{R_k^{i+1,j}} = \boldsymbol{R_{k+1}^{ij}}$ for $k = -p+1, \ldots p-2$
   (g) Compute $\boldsymbol{R_{p-1}^{i+1,j}}$
   (h) Iterate until end of current column
3. Iterate until convergence

is minimized by:

$$\hat{l}_{rs} = \frac{\phi'(x_r - x_s)}{2(x_r - x_s)} \tag{1.75}$$

The duality principle is important in that Blake (Blake and Zisserman 1987) shows that introducing a second variable, in order to perform the minimization above, is equivalent to explicitly introducing a line field in the model. Blake demonstrates this equivalence in the case of a Gaussian function V_c, while the extension of this result to general convex functions and done by Geman (Geman and Reynolds 1992).

By using these results we obtain algorithms that cost up to four times less than conjugate gradient algorithms as well as SSUAs proposed by Bouman and Sauer

(Sauer and Bouman 1993). Furthermore, visually the resulting images are good with the homogeneous regions well separated by abrupt transitions.

1.5 Parameter Estimation

Several methods of choosing parameters have been proposed in the recent literature. (Galatsanos and Katsaggelos 1992) and (Thompson, Brown, Kay and Titterington 1991) contain comparisons of the various methods available. It is important to note that algorithms which estimate parameters have only been found in the case when a quadratic criterion is being minimized. Work is still being done for the parameter estimation problem for Markov models however at this point none exist. We note that even in the quadratic case, the estimated parameters are not always accurate enough to be used in practice so some adjustments must be made.

In the quadratic case the solution to Equation 1.7 is given explicitly by:

$$\hat{\boldsymbol{x}}(\mu, \boldsymbol{y}) = (\boldsymbol{H}^t\boldsymbol{H} + \mu\boldsymbol{Q}^t\boldsymbol{Q})^{-1}\boldsymbol{H}^t\boldsymbol{y} \tag{1.76}$$

(which is Equation 1.22 restated for convenience). Consequently, we consider the problem of estimating μ in the above expression. This parameter estimation problem has been frequently considered in the recent literature and several methods have appeared. Several ways of estimating the parameter have appeared in the literature including the total predicted mean squared error TPMSE given by:

$$\mu_{TPMSE} = \arg\min_{\mu} E\|\boldsymbol{H}\boldsymbol{x} - \hat{\boldsymbol{y}}(\mu)\|^2 \tag{1.77}$$

However this estimate has the undesirable property that it is a function of the unknown $\boldsymbol{x}$ and in order to calculate the expectation we will require the variance of the noise r_n. Furthermore minimizing this criterion is a non-trivial exercise. Other possibilities include the constrained least squares approach and the equivalent degrees of freedom approach both discussed in (Galatsanos and Katsaggelos 1992). However

these also suffer from the drawback that we require the variance of the noise r_n. The L-curve method is another approach which has been used (Brooks, Maratos and MacLeod 1993). It consists of evaluating the ratio between the logarithm of $\|\hat{\boldsymbol{x}}\|^{\frac{1}{2}}$ and the logarithm of $\|\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}\|^{\frac{1}{2}}$ which is the magnitude of the prediction error. This yields an "L" shaped curve and the value of μ yielding the corner point is the parameter which is chosen. The problem with this approach is that it is a heuristic approach and unlike with cross validation, as we will see, no strong convergence results exist.

Two other approaches, which are more interesting, are the cross validation (CV) approach and the maximum likelihood (ML) method, both of which do not require the noise variance. Consequently, we will consider these more closely since in our case, we do not have the variance of the noise. A weighted version of the CV algorithm has been developed and is known as generalized cross validation (GCV). However in the case of an invariant point spread function GCV is equivalent to CV.

The CV algorithm consists of removing a sample from $\boldsymbol{y}$ and then predicting its value from the rest of the data. The main idea is that minimizing this estimation error over all samples will yield a reasonable choice of parameters. It was also shown by Li in (Li, Held, Curlander and Wu 1985, Li 1986) that for GCV

$$\lim_{M\to\infty} [\frac{\|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{H}\hat{\boldsymbol{x}}(\mu_{GCV}, \boldsymbol{y})\|^2}{\inf_{\mu>0}\|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{H}\hat{\boldsymbol{x}}(\mu, \boldsymbol{y})\|^2}] = 1 \tag{1.78}$$

where $\hat{\boldsymbol{x}}(\mu, \boldsymbol{y})$ is the estimate of $\boldsymbol{x}$ using an arbitrary $\mu > 0$and M is the number of pixels. In words, we have that for an infinite number of pixels, the μ predicted by GCV minimizes the prediction error $\|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{H}\hat{\boldsymbol{x}}(\mu, \boldsymbol{y})\|^2$. Although we have a finite number of pixels this result still strongly motivates the use of GCV.

Since for our case GCV and CV are equivalent we ignore the complications introduced in presenting GCV since they lead to an identical result. The CV criterion is

given by:

$$CV(\mu) = \sum_k [y_k - \hat{y}_k^{-k}]^2 \tag{1.79}$$

where $\hat{y}_k^{-k}$ denotes the kth sample of $\boldsymbol{y}$ predicted from the image restored from all the data except the kth pixel. In order to calculate this quantity, we must first calculate $\hat{\boldsymbol{x}}(\mu, \boldsymbol{y}^{(-k)})$ which is $\boldsymbol{x}$ estimated from all the samples of $\boldsymbol{y}$ except sample k. To calculate this, we apply the well-known matrix inversion lemma to

$$\hat{\boldsymbol{x}}(\mu, \boldsymbol{y}^{(-k)}) = (\boldsymbol{H}^t\boldsymbol{H} + \mu\boldsymbol{Q} - \boldsymbol{h}_k\boldsymbol{h}_k^t)^{-1}(\boldsymbol{H}^t\boldsymbol{y} - \boldsymbol{h}_k y_k - \boldsymbol{H}^t\boldsymbol{H}\hat{\boldsymbol{x}}_\infty) \tag{1.80}$$

to obtain the prediction error

$$e^{(-k)} = y_k - \boldsymbol{h}_k^t\hat{\boldsymbol{x}}(\mu, \boldsymbol{z}^{(-k)}) = \frac{u_k}{1 - a_{kk}} \tag{1.81}$$

where u_k is the kth residual error

$$u_k = y_k - \boldsymbol{h}_k\hat{\boldsymbol{x}}(\mu, \boldsymbol{y}) \tag{1.82}$$

and where a_{kk} denoted the diagonal elements of the matrix $\boldsymbol{A}(\mu)$ defined by

$$\boldsymbol{A}(\mu) = \boldsymbol{H}(\boldsymbol{H}^t\boldsymbol{H} + \mu\boldsymbol{Q})^{-1}\boldsymbol{H}^t \tag{1.83}$$

Using this notation the CV criterion can be written as

$$\boldsymbol{P}(\mu) = \frac{1}{M}\sum_{k=1}^{M}(\frac{u_k}{1 - a_{kk}})^2 \tag{1.84}$$

Minimization of the function $\boldsymbol{P}(\mu)$ in 1.84 is a complicated nonlinear problem which

must be solved iteratively. Most often the range of variation of μ is sampled, the corresponding values of the CV criterion are evaluated, and the resulting minimum is selected. In (Fortier, Demoment and Goussard 1993a) it is shown how the Chandresakhar filter (derived in Section 2.2) can be used to efficiently calculate this criterion.

Another method is the Maximum Likelihood (ML) method which is only applicable when the stochastic framework is adopted. This method is also particularly useful as it does not require the object $\boldsymbol{x}$ to be estimated . Here the idea is to choose the value of μ which maximizes $p(\boldsymbol{y}|\mu)$. In order to calculate this probability we first note that the joint probability density function of the observations is

$$p(\boldsymbol{y}|\mu,\sigma_b^2) = (2\pi)^{-M/2}|\boldsymbol{R}_y|^{-1/2}\exp\left[-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{H}\hat{\boldsymbol{x}}_\infty)^t\boldsymbol{R}_z^{-1}(\boldsymbol{y}-\boldsymbol{H}\hat{\boldsymbol{x}}_\infty)\right] \quad (1.85)$$

where

$$\boldsymbol{R}_z = \boldsymbol{H}\boldsymbol{R}_x\boldsymbol{H}^t + \boldsymbol{R}_b = \sigma_b^2(\boldsymbol{I}-\boldsymbol{A}(\mu))^{-1} \quad (1.86)$$

maximizing with respect to σ_b^2 yields

$$\sigma_b^2 = \frac{1}{\boldsymbol{M}}(\boldsymbol{y}-\boldsymbol{H}\hat{\boldsymbol{x}}_\infty)^t(\boldsymbol{I}-\boldsymbol{A}(\mu))(\boldsymbol{y}-\boldsymbol{H}\hat{\boldsymbol{x}}_\infty) \quad (1.87)$$

substituting into 1.85 we get

$$p(\boldsymbol{y}|\mu,\sigma_b^2) = (2\pi)^{-M/2}\left[\frac{1}{\boldsymbol{M}}(\boldsymbol{y}-\boldsymbol{H}\hat{\boldsymbol{x}}_\infty)^t(\boldsymbol{I}-\boldsymbol{A}(\mu))(\boldsymbol{y}-\boldsymbol{H}\hat{\boldsymbol{x}}_\infty)\right]^{-M/2} |\boldsymbol{I}-\boldsymbol{A}(\mu)|^{1/2}\exp(-\frac{M}{2}) \quad (1.88)$$

Maximization of the above expression with respect to μ is equivalent to minimizing

$$\frac{(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_\infty)^t(\boldsymbol{I} - \boldsymbol{A}(\mu))(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_\infty)}{|\boldsymbol{I} - \boldsymbol{A}(\mu)|^{1/M}} \tag{1.89}$$

This is often referred to as the generalized maximum likelihood criteria (GML) developed by Wahba in the context of spline smoothing problems (Wahba 1985). Once again minimization of this expression is a complicated nonlinear problem. In (Fortier et al. 1993a) the Chandresakhar filter is used to efficiently compute this criterion. Another method based on Expectation Maximization was also proposed in (Lagendijk, Biemond and Boekee 1989). Simulation studies in (Fortier et al. 1993a, Thompson et al. 1991) indicate that the ML method yields a relatively accurate estimate of the true parameter. The CV estimate however tends to be lower than the true parameter and consequently undersmoothing is observed in the resulting images.

In the article included in Section 3 a Cross Validation based method is presented in the context of the constrained Wiener filter. We adopt this method because it depends only on the data. Furthermore, in the case of parameter estimation for the Wiener filter, it is possible to obtain a computationally efficient solution.

1.6 Critique

We have considered two families of solutions to the image restoration problem in this literature survey: solutions arising from quadratic criteria and those arising from Markov Random Fields. We recall that our main objective was to obtain a suitable compromise between suppression of the noise and reduction of the blur. We are also interested in a reasonably rapid algorithm and one that can take into account the correlation between slices which corresponds to a 3D implementation. Finally we were interested in algorithms in which parameters can be selected automatically.

While adopting a MRF approach does yield smooth homogeneous regions separated by abrupt transitions, we note that these algorithms are more expensive. Furthermore automatic selection of parameters is not possible at present with Markov Random Fields.

In what follows, we will adopt a quadratic criterion since, as we have seen, this leads to explicit solutions. Rapid calculation of these solutions is made possible by the availability of algorithms based on the Kalman filter and conjugate gradient. Another advantage of using a quadratic criterion is the fact that parameters can be selected automatically by using such methods as ML and CV.

However, the algorithms described in this literature survey which minimize a quadratic criterion are insufficient for our application. The conjugate gradient method cannot be extended to three dimensions because of the large size of the resulting matrices. Furthermore this method treats the image globally and consequently cannot take into account local characteristics such as the presence of an edge or a homogeneous region.

Similarly, the Kalman filter also cannot be extended to three dimensions because of the large matrices which arise. In Chapter 2.1 we examine methods which allow us to

reduce the computational complexity without significantly effecting the performance, however these methods still cannot be extended to 3D. Furthermore, the Kalman filter we develop minimizes a global criteria and cannot be tuned to local characteristics in the image.

In order to compensate for these deficiencies, we will introduce adaptive methods which will attempt to recover boundaries at transition points and smooth homogeneous regions. In particular, we will develop a filtering approach based on the constrained Wiener filter in Section 3 which will allow the data to be processed locally. We will see that the constraining of the filter leads to a reduced computational load which allows the algorithm to be extended to three dimensions. Furthermore, we develop an algorithm based on CV to estimate the parameter for this filter.

Chapter 2

Recursive Filtering Methods

This chapter describes recursive filtering methods which solve Equation 1.7 with J_1 and J_2 chosen to be quadratics. The methods of this class that were presented in the previous chapter are based on a causal auto-regressive (AR) model. This causality, which is not related to any intrinsic property of the image, introduces sub-optimalities. In order to avoid this disadvantage, we adopt a degenerate model in which the entire state $\boldsymbol{x}$ is estimated at each iteration. Using this model we obtain a Kalman filter which yields the same estimate as the Kalman smoother. Therefore, we avoid another sub-optimality of the AR-based methods, *i.e.*, substitution of a Kalman filter for a Kalman smoother. However this method yields a high volume of calculations so we develop factorization methods which greatly reduce the computational load. We also develop square root factorizations which are used to improve the numerical stability.

2.1 Kalman Filtering

In this section recursive methods based on the Kalman filter are developed for solving the image restoration problem. We adopt recursive methods as these lead to a

reduced computational burden as we will demonstrate when we consider factorization methods. As indicated in the literature survey, the Kalman filter gives the solution to Equation 1.7 with J_1 and J_2 chosen to be quadratics.

Before presenting the Kalman filter adapted to the problem of image restoration, a more general problem from the stochastic control field is considered. We subsequently discuss how this model must be adapted to be used in the context of image restoration.

The following system of equations been extensively studied in the stochastic control domain:

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k \boldsymbol{x}_k + \boldsymbol{G}_k \boldsymbol{u}_k \tag{2.1}$$

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k \tag{2.2}$$

where $\boldsymbol{x}_k$ is the state of the system at time k, $\boldsymbol{y}_k$ is the observed vector, $\boldsymbol{H}_k$, $\boldsymbol{F}_k$, and $\boldsymbol{G}_k$ are known matrices at all instants k, and $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ are sequences of Gaussian white noise independent of each other and independent of $\boldsymbol{x}_k$ such that we have:

$$E[\boldsymbol{u}_k] = E[\boldsymbol{v}_k] = 0 \tag{2.3}$$

$$E[\boldsymbol{u}_m \boldsymbol{u}_k^t] = \boldsymbol{Q}_u \delta_{mk} \tag{2.4}$$

$$E[\boldsymbol{v}_m \boldsymbol{v}_k^t] = \boldsymbol{R} \delta_{mk} \tag{2.5}$$

$$E[\boldsymbol{v}_m \boldsymbol{u}_k^t] = 0 \tag{2.6}$$

where $\boldsymbol{Q}_u$ and $\boldsymbol{R}$ are covariance matrices and δ is the Kronecker delta function.

Several types of *a priori* knowledge are introduced into the system. Firstly, the vector to be restored $\boldsymbol{x}$ is assumed to contain samples which are distributed as Gaus-

sians with:

$$E[\boldsymbol{x}_0] = \underline{\boldsymbol{m}}_0 \tag{2.7}$$

$$E[\boldsymbol{x}_0 \boldsymbol{x}_0^t] = Cov[\boldsymbol{x}_0] = \boldsymbol{P}_0 \tag{2.8}$$

$\boldsymbol{m}$ is a parameter which is the average value of the vector $\boldsymbol{x}$ and $\boldsymbol{P}_0$ is a covariance matrix. Furthermore, the noise is assumed Gaussian with mean equal to 0 and covariance matrices $\boldsymbol{Q}_u$ and $\boldsymbol{R}$ also supply *a priori* knowledge to the system. $\boldsymbol{P}_0$, $\boldsymbol{Q}_u$, $\boldsymbol{R}$ and $\underline{\boldsymbol{x}}_0$ are all parameters which must be supplied to the system based on *a priori* knowledge of the signal $\boldsymbol{x}$ and the noise processes $\boldsymbol{u}$ and $\boldsymbol{v}$.

Our goal is to find an expression for $E[\boldsymbol{x}_k | y_1 \ldots y_N]$. We recall from the literature survey that in the Gaussian case this equals the MAP estimate as the expected value occurs at the maximum. In words, the aim is to estimate the state of the system given all the observations up to and including the present observation. To obtain such an expression we begin by developing the associated probability density function by using Bayes' rule as follows:

$$f(\boldsymbol{x}_{k+1} | \tilde{\boldsymbol{y}}_{k+1}) = \frac{f(\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1} | \tilde{\boldsymbol{y}}_k)}{f(\boldsymbol{y}_{k+1} | \tilde{\boldsymbol{y}}_k)} \tag{2.9}$$

$$= \frac{f(\boldsymbol{y}_{k+1} | \boldsymbol{x}_{k+1})}{f(\boldsymbol{y}_{k+1} | \tilde{\boldsymbol{y}}_k)} f(\boldsymbol{x}_{k+1} | \tilde{\boldsymbol{y}}_k) \tag{2.10}$$

where $\tilde{\boldsymbol{y}}_k$ represents $\boldsymbol{y}_1, \boldsymbol{y}_2 \ldots \boldsymbol{y}_k$. Since all the probability densities in 2.10 are Gaussians due to the hypotheses made concerning the noise and vector $\boldsymbol{x}$, we need only find the mean and the covariance of all the quantities on the right hand side of 2.10 in order to determine the distribution we seek, namely $f(\boldsymbol{x}_{k+1} | \tilde{\boldsymbol{y}}_{k+1})$.

For the sake of clarity, the following notation is introduced:

$$\hat{\boldsymbol{x}}_{k|k} \triangleq E[\boldsymbol{x}_k|\tilde{\boldsymbol{y}}_k] \tag{2.11}$$

$$\boldsymbol{P}_{k|k} \triangleq Cov[\boldsymbol{x}_k|\tilde{\boldsymbol{y}}_k] \tag{2.12}$$

Using this notation and a straightforward application of the system model equations 2.1 and 2.2 yields the following expressions for the sought after means and covariances:

$$E[\boldsymbol{x}_{k+1}|\tilde{\boldsymbol{y}}_k] = \boldsymbol{F}_k\hat{\boldsymbol{x}}_{k|k} \tag{2.13}$$

$$Cov[\boldsymbol{x}_{k+1}|\tilde{\boldsymbol{y}}_k] = \boldsymbol{F}_k\boldsymbol{P}_{k|k}\boldsymbol{F}_k^t + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^t \tag{2.14}$$

$$\triangleq \boldsymbol{P}_{k+1|k}$$

$$E[\boldsymbol{y}_{k+1}|\tilde{\boldsymbol{y}}_k] = \boldsymbol{H}_k\boldsymbol{F}_k\hat{\boldsymbol{x}}_{k|k} \tag{2.15}$$

$$Cov[\boldsymbol{y}_{k+1}|\tilde{\boldsymbol{y}}_k] = \boldsymbol{H}_k\boldsymbol{P}_{k+1|k+1}\boldsymbol{H}_k^t \tag{2.16}$$

$$E[\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1}] = \boldsymbol{H}_{k+1}\boldsymbol{x}_{k+1} \tag{2.17}$$

$$Cov[\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1}] = \boldsymbol{H}_{k+1}\boldsymbol{P}_{k+1|k+1}\boldsymbol{H}_{k+1}^t + \boldsymbol{R}_{k+1}. \tag{2.18}$$

The Gaussian probability density on the left hand side of 2.10 can be expressed in the well known form as follows:

$$f(\boldsymbol{x}_{k+1}|\tilde{\boldsymbol{y}}_{k+1}) = \boldsymbol{a}\exp\left(-\frac{1}{2}\right)[\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k+1}]^t\boldsymbol{P}_{k+1|k+1}^{-1}[\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k+1}] \tag{2.19}$$

Substituting the expressions for the means and covariances into 2.10 and matching

the means and covariances yields the following expressions for the Kalman filter:

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \boldsymbol{F}_k\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{P}_{k+1|k}\boldsymbol{H}_k^t[\boldsymbol{H}_k\boldsymbol{P}_{k+1|k}\boldsymbol{H}_k^t + \boldsymbol{R}_k]^{-1}[\boldsymbol{y}_{k+1} - \boldsymbol{H}_k\boldsymbol{F}_k\hat{\boldsymbol{x}}_{k|k}] \quad (2.20)$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{F}_k\boldsymbol{P}_{k|k}\boldsymbol{F}_k^t + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^t \quad (2.21)$$

$$\boldsymbol{P}_{k+1|k+1}^{-1} = \boldsymbol{P}_{k+1|k}^{-1} + \boldsymbol{H}_k^t\boldsymbol{R}_k^{-1}\boldsymbol{H}_k \quad (2.22)$$

These expressions can be rearranged as follows to yield a form which separates the two stages of prediction and correction:

Prediction:

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{F}_k\hat{\boldsymbol{x}}_{k|k} \quad (2.23)$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{F}_k\boldsymbol{P}_{k|k}\boldsymbol{F}_k^t + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^t \quad (2.24)$$

Correction:

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{K}_{k+1}[\boldsymbol{y}_{k+1} - \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k+1|k}] \quad (2.25)$$

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1|k}\boldsymbol{H}_{k+1}^t\boldsymbol{E}_{k+1}^{-1} \quad (2.26)$$

$$\boldsymbol{E}_{k+1} = \boldsymbol{R}_{k+1} + \boldsymbol{H}_{k+1}\boldsymbol{P}_{k+1|k}\boldsymbol{H}_{k+1}^t \quad (2.27)$$

$$\boldsymbol{P}_{k+1|k+1} = [\boldsymbol{I} - \boldsymbol{K}_{k+1}\boldsymbol{H}_{k+1}]\boldsymbol{P}_{k+1|k} \quad (2.28)$$

In the prediction stage, $\boldsymbol{x}_{k+1}$ is estimated from the observations up to time k. In the correction stage the estimate is adjusted to correct for the new observation at time $k+1$. The algorithm is initialized as follows:

$$\boldsymbol{P}_{0|-1} = \text{identity matrix (no correlation) or Toeplitz (with correlation)} \quad (2.29)$$

$$\hat{\boldsymbol{x}}_{0|-1} = \boldsymbol{E}[x] = \boldsymbol{m} \quad (2.30)$$

Equations 2.23-2.28 along with the initializations constitute the standard Kalman filter. We note that most expensive computations are the updating of the covariance matrix $\boldsymbol{P}$ in equations 2.24 and 2.27, and the inversion of the matrix $\boldsymbol{E}$. The size of $\boldsymbol{P}$ is $N \times N$ where N is the number of observations. The number of elements in the matrix $\boldsymbol{E}$ is equal to the size of the observation squared.

In Equation 2.1, we note that the state is changing at each iteration. Consequently, this general model is not appropriate for the image restoration problem in which the image to be estimated is constant. In the literature survey we restricted ourselves to auto-regressive models in which the matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ remain constant and take the forms specified in Equations 1.49 and 1.50. Also $\boldsymbol{x_k}$ is defined by Equation 1.50 and $\boldsymbol{u_k}$ becomes a scalar which we denote u_k. Consequently the matrix $\boldsymbol{Q}$ in 2.4 also becomes a scalar. However in the literature survey we noted that several sub-optimalities were introduced as a result of using an auto-regressive model in 2D.

In order to eliminate these sub-optimalities we will adopt a degenerate model in which the state remains constant and is equal to the entire image to be estimated. The adapted model is given by equations 2.1 and 2.2.

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k = const. \tag{2.31}$$

$$y_k = \boldsymbol{h}_k^t \boldsymbol{x}_k + v_k \tag{2.32}$$

$$\tag{2.33}$$

Here $\boldsymbol{x}$ is the entire image to be restored which is constant. It is a column vector formed by concatenating the rows of the image. Unlike the RUKF presented in the literature survey, the entire image will be estimated with each observation. Consequently, at the end of the filtering process we will have calculated $E[\boldsymbol{x}_k|\boldsymbol{y}_1, \boldsymbol{y}_2 \cdots \boldsymbol{y}_k]$. This corresponds to the Kalman smoother. In order to obtain this estimate with the

RUKF two passes would have to be made with some modifications to the basic filter. By adopting our model, we omit the need for two passes which saves on computation without introducing any sub-optimalities.

We also note that unlike the basic model, here we have chosen to use a scalar observation. This immediately eliminates the inversion of the matrix $\boldsymbol{E}$ which now becomes a scalar. thus reducing our volume of calculations.

Although the PSF is assumed shift invariant over the image, the quantity $\boldsymbol{h}_k$ is not. This results from the fact that by concatenating the rows of the image into the vector $\boldsymbol{x}$. we must modify the $\boldsymbol{h}$ at the edges of the image by replacing samples of the PSF with the zeros so that we do not falsely convolve the left edge of the image with the right edge. The filter equations for this model are obtained by substituting $\boldsymbol{F}_k = \boldsymbol{I}$ and $\boldsymbol{u}_k = 0$ in 2.1. Making these substitutions yields the following:

$$\hat{\boldsymbol{x}}_{k+1|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{k}_k e_k^{-1}[\boldsymbol{y}_{k+1} - \boldsymbol{H}_k \hat{\boldsymbol{x}}_{k+1|k}] \tag{2.34}$$

$$e_k = \boldsymbol{h}_k^t \boldsymbol{P}_{k|k-1} \boldsymbol{h}_k + r_k \tag{2.35}$$

$$\boldsymbol{k}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{h}_k \tag{2.36}$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{k}_k e_k^{-1} \boldsymbol{k}_k^t \tag{2.37}$$

These equations are initialized as before with

$\boldsymbol{P}_{0|-1}$ = identity matrix (no correlation) or Toeplitz-block-Toeplitz (with correlation)

$\hat{\boldsymbol{x}}_{0|-1} = \boldsymbol{E}[x]$

The Toeplitz-block-Toeplitz structure in the case with correlation is a direct consequence of the fact that the rows of the image are stacked one after another in the vector $\boldsymbol{x}$. Several observations must be made on the adapted Kalman filter ex-

pressed in equations 2.34-2.37. We immediately note that no matrix inversions are involved. Nevertheless, the Kalman filter is still computationally expensive. For an image 200 × 200 pixels, the covariance matrix $\boldsymbol{P}$ is 40000 × 40000. Consequently, iterating over the 40000 pixels is not feasible. In order to reduce the computational load, the fast Kalman filter can be used (Demoment 1987).

2.2 Chandrasekhar Filtering

As indicated in the previous section, the problem with the standard Kalman filter adapted to the image restoration problem is the computational burden associated with updating the covariance matrix. Rather than update the matrix, the Chandrasekhar equations reduce the computational load of the system by updating increments. If the *a priori* matrix $\boldsymbol{P}_0$ is well structured, for example Toeplitz or identity, it is possible to factor the initial increment and develop a new algorithm which propagates the factors. The major computational saving occurs when these factors are propagated as opposed to the actual increments.

This approach can be applied to the general system given in equations 2.1 and 2.2 provided that the matrices $\boldsymbol{F}$,$\boldsymbol{G}$ and $\boldsymbol{H}$ remain constant, but here we immediately consider the special case adapted to image restoration.

2.2.1 Optimal Filters

We first demonstrate how the system can be factorized, and then, subsequently, we will consider how further computational savings can be made if we are willing to accept a small sub-optimality.

We first introduce the new system model:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k = const. \tag{2.38}$$

$$\boldsymbol{y}_k = \boldsymbol{h}_k^t \boldsymbol{x}_k + \boldsymbol{v}_k \tag{2.39}$$

$$\boldsymbol{h}_{k+1} = \boldsymbol{D}\boldsymbol{h}_k \tag{2.40}$$

where $\boldsymbol{D}$ is a shifting matrix with ones on the first sub-diagonal and zeros everywhere else. This matrix shifts the vector $\boldsymbol{h}$ one sample to the right. This is the same as the model given by 2.31 and 2.32 except that we insist here that $\boldsymbol{h}$ is shift-invariant. Consequently, the vector $\boldsymbol{x}$ must be padded with zeros between each row of the image that is inserted. The number of zeros inserted between each row is equal to the width of the PSF. This is necessary so that pixels on the right edge of the image are not convolved with those on the left edge of the image. The resulting filter is sub-optimal as we have introduced new observations into the system. This same sub-optimality occurs in the RUKF where zeros must also be added to properly isolate the right side of the image from the left. However, the effect is minimal since although the filter updates the whole image at each iteration, we find that the nonzero elements of the filter occur in the region of the PSF. Consequently, when an observation is made at the edge of the image only the edge pixels in the vicinity are effected. For our purposes this effect can be neglected since the interesting part of the image is in the center, well removed from the edge. We will see that the by adopting a shift-invariant PSF leads to a system which can be factored in a way which will considerably reduce the computational complexity.

We introduce the following notation which will prove useful in our developments:

$$\begin{aligned} \boldsymbol{g}_k &= \boldsymbol{k}_k e_k^{-1} \text{where } \boldsymbol{g}_k \text{ is the generalized gain} && (2.41)\\ \delta \boldsymbol{g}_k &= \boldsymbol{g}_k - \boldsymbol{D}\boldsymbol{g}_{k-1} && (2.42)\\ \delta e_k &= e_k - e_{k-1} && (2.43)\\ \delta \boldsymbol{P}_k &= \boldsymbol{P}_{k|k-1} - \boldsymbol{D}\boldsymbol{P}_{k-1|k-2}\boldsymbol{D}^t && (2.44) \end{aligned}$$

We now develop the filter using these definitions. Using 2.37 we obtain:

$$\begin{aligned} \delta \boldsymbol{P}_{k+1} &= \boldsymbol{P}_{k|k-1} - \boldsymbol{k}_k e_k^{-1} \boldsymbol{k}_k^t - \boldsymbol{D}[\boldsymbol{P}_{k-1|k-2} - \boldsymbol{k}_{k-1} e_{k-1}^{-1} \boldsymbol{k}_{k-1}^t]\boldsymbol{D}^t && (2.45)\\ &= \delta \boldsymbol{P}_k + \boldsymbol{D}\boldsymbol{k}_{k-1} e_{k-1}^{-1} \boldsymbol{k}_{k-1}^t \boldsymbol{D}^t - \boldsymbol{k}_k e_k^{-1} \boldsymbol{k}_k^t && (2.46) \end{aligned}$$

also 2.35 yields:

$$\begin{aligned} e_k - e_{k-1} &= \boldsymbol{h}_k^t \boldsymbol{P}_{k|k-1} \boldsymbol{h}_k - \boldsymbol{h}_k^t \boldsymbol{D}\boldsymbol{P}_{k|k-1}\boldsymbol{D}^t \boldsymbol{h}_k && (2.47)\\ &= \boldsymbol{h}_k^T \delta \boldsymbol{P} \boldsymbol{h}_k && (2.48) \end{aligned}$$

so that

$$\begin{aligned} \boldsymbol{g}_k &= \boldsymbol{P}_{k|k-1}\boldsymbol{h}_k e_k^{-1}\\ &= \delta \boldsymbol{P}_{k-1}\boldsymbol{h}_k e_k^{-1} + \boldsymbol{D}\boldsymbol{P}_{k-1}\boldsymbol{D}^t \boldsymbol{h}_k e_k^{-1}\\ &= \delta \boldsymbol{P}_{k-1}\boldsymbol{h}_k e_k^{-1} + \boldsymbol{D}\boldsymbol{P}_{k-1}\boldsymbol{h}_{k-1} e_k^{-1}\\ &= [\delta \boldsymbol{P}_k \boldsymbol{h}_k + \boldsymbol{D}\boldsymbol{g}_{k-1} e_{k-1}] e_k^{-1} \end{aligned}$$

We also have that

$$\delta \boldsymbol{g}_k = (\boldsymbol{I} - \boldsymbol{D}\boldsymbol{g}_k \boldsymbol{h}_k^t)\delta \boldsymbol{P}_k \boldsymbol{h}_k e_k^{-1} \quad (2.49)$$

using these results allows us to express 2.45 as follows:

$$\begin{aligned} \delta \boldsymbol{P}_{k+1} &= \delta \boldsymbol{P}_k - [\boldsymbol{D}\boldsymbol{g}_{k-1} + \delta \boldsymbol{g}_k] e_k [\boldsymbol{D}\boldsymbol{g}_{k-1} + \delta \boldsymbol{g}_k]^t \\ &+ \boldsymbol{D}\boldsymbol{g}_{k-1} [e_k - \boldsymbol{H}\delta \boldsymbol{P}_k \boldsymbol{H}^t] \boldsymbol{g}_{k-1} \boldsymbol{D}^t \end{aligned} \tag{2.50}$$

developing the preceding equation and using the above expressions allows us to factor $\boldsymbol{P}_{k+1}$ as follows:

$$\delta \boldsymbol{P}_{k+1} = [\boldsymbol{I} - \boldsymbol{g}_k \boldsymbol{h}^t][\delta \boldsymbol{P}_k + \delta \boldsymbol{P}_k \boldsymbol{h} e_{k-1}^{-1} \boldsymbol{h}^t \delta \boldsymbol{P}_k][\boldsymbol{I} - \boldsymbol{g}_k \boldsymbol{h}^t]^t \tag{2.51}$$

If we now assume that $\delta \boldsymbol{P}_1$ can be factored as follows:

$$\delta \boldsymbol{P}_1 = \boldsymbol{Y}_0 \boldsymbol{M}_0 \boldsymbol{Y}_0^t \tag{2.52}$$

we obtain the recursive set of equations:

$$\delta \boldsymbol{P}_{k+1} = \boldsymbol{Y}_k \boldsymbol{M}_k \boldsymbol{Y}_k^t \tag{2.53}$$

$$\hat{\boldsymbol{x}}_{k+1|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{k}_k e_k^{-1} [\boldsymbol{y}_{k+1} - \boldsymbol{h}_k \hat{\boldsymbol{x}}_{k+1|k}] \tag{2.54}$$

$$e_{k+1} = e_k + \boldsymbol{h}_{k+1}^t \boldsymbol{Y}_k \boldsymbol{M}_k \boldsymbol{Y}_k^t \boldsymbol{h}_{k+1} \tag{2.55}$$

$$\boldsymbol{k}_{k+1} = \boldsymbol{D}\boldsymbol{k}_k + \boldsymbol{Y}_k \boldsymbol{M}_k \boldsymbol{Y}_k^t \boldsymbol{h}_{k+1} \tag{2.56}$$

$$\boldsymbol{Y}_k = [\boldsymbol{I} - \boldsymbol{g}_k \boldsymbol{h}^t] \boldsymbol{Y}_{k-1} \tag{2.57}$$

$$\boldsymbol{M}_k = \boldsymbol{M}_{k-1} \boldsymbol{Y}_{k-1}^t \boldsymbol{h} e_k^{-1} \boldsymbol{h}^t \boldsymbol{Y}_{k-1} \boldsymbol{M}_{k-1}^t \tag{2.58}$$

In order to reduce the computational load, auxiliary quantities can be introduced as

in (Fortier, Demoment and Goussard 1993b) to yield the following algorithm:

$$\boldsymbol{v}_k^t = \boldsymbol{h}_k^t \boldsymbol{Y}_{k-1} \tag{2.59}$$

$$\boldsymbol{w}_k^t = \boldsymbol{v}_k^t \boldsymbol{M}_{k-1} \tag{2.60}$$

$$e_k = e_{k-1} + \boldsymbol{v}_k^t \boldsymbol{w}_k \tag{2.61}$$

$$\boldsymbol{k}_k = \boldsymbol{D}\boldsymbol{k}_{k-1} + \boldsymbol{Y}_{k-1}\boldsymbol{w}_k \tag{2.62}$$

$$\hat{\boldsymbol{x}}_{k|k-1} = \hat{\boldsymbol{x}}_{k-1|k-2} + \boldsymbol{k}_k e_k^{-1}[\boldsymbol{y}_{k+1} - \boldsymbol{h}_k \hat{\boldsymbol{x}}_{k+1|k}] \tag{2.63}$$

$$\boldsymbol{M}_k = \boldsymbol{M}_{k-1} + \boldsymbol{w}_k e_{k-1} \boldsymbol{w}_k^t \tag{2.64}$$

$$\boldsymbol{Y}_k = \boldsymbol{Y}_{k-1} - \boldsymbol{k}_k e_{k-1} \boldsymbol{v}_k^t \tag{2.65}$$

The algorithm is initialized by factoring the first increment $\delta \boldsymbol{P}_0$ in terms of the matrices $\boldsymbol{M}$ and $\boldsymbol{Y}$. In order to factor the system in this way, $\boldsymbol{P}_0$ must be Toeplitz and not Toeplitz block Toeplitz as with the standard filter. $\boldsymbol{P}_0$ can be chosen as a Toeplitz matrix since in our new vector $\boldsymbol{x}$ we have isolated neighboring rows by padding with zeros. The Toeplitz matrix is the key to the factorization of the system since it yields an increment which can be factored in terms of the low rank matrix

$\boldsymbol{M}$.

$$
\begin{aligned}
\boldsymbol{P}_0 &= \boldsymbol{R}_x \text{ Toeplitz matrix} \\
&= \boldsymbol{r}\boldsymbol{r}^t && (2.66) \\
\boldsymbol{r} &= \begin{bmatrix} \gamma_{11} & \dots & \gamma_{1m} & 0 & \dots & \dots & \gamma_{n1} & \dots & \gamma_{nm} & 0 \end{bmatrix} && (2.67) \\
\hat{\boldsymbol{x}}_0 &= 0 \\
e_1 &= r_b + \boldsymbol{h}_1 \boldsymbol{P}_0 \boldsymbol{h}^t \\
\boldsymbol{k}_1 &= \boldsymbol{P}_{0|-1}\boldsymbol{h}_1 \\
\delta \boldsymbol{P}_1 &= \boldsymbol{P}_1 - \boldsymbol{D}\boldsymbol{P}_0\boldsymbol{D}^t \\
&= \begin{bmatrix} 1 & | & \nu^t & | & 0^t \\ - & - & - & - & - \\ \nu & | & & & \\ - & | & & 0 & \\ 0 & | & & & \end{bmatrix} - \boldsymbol{k}_1 e_1^{-1} \boldsymbol{k}_1^t
\end{aligned}
$$

where ν contains the values of $E[\boldsymbol{X}\boldsymbol{X}^t]$ which can be chosen with or without correlation. n and m are the correlation distance in the x and y directions respectively. $\delta \boldsymbol{P}$ can now be factorized as follows in order to be used in the Chandresakhar equations.

$$
\begin{aligned}
\boldsymbol{Y}_1 &= \begin{bmatrix} & | & 1 & | & 0 \\ \boldsymbol{k}_1 e_1^{-1/2} & | & 0 & | & \nu \\ & | & 0 & | & \end{bmatrix} \\
\boldsymbol{M}_1 &= \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \\
\delta \boldsymbol{P}_0 &= \boldsymbol{Y}_1 \boldsymbol{M}_1 \boldsymbol{Y}_1^t
\end{aligned}
$$

In practice we find that when $E[\boldsymbol{X}\boldsymbol{X}^t]$ is chosen with correlation the resulting algorithms suffer from numerical instabilities. The problem lies in the fact that in the algorithm, positive quantities of the same magnitude are being subtracted from each other. However, due to the finite precision of the computer, results which should be positive, occasionally turn negative (Verhaegen and Van Dooren 1986).

2.2.2 Sub-optimal Filters

A key observation which greatly reduces the computational complexity is the observation that the quantity $\boldsymbol{k}_k e_k^{-1}$, which we refer to as the gain vector, tends to a value which is shift invariant. We further observe that the calculation of this vector is independent of the observed image. The gain vector depends only on the *a priori* knowledge contained in the matrix $\delta \boldsymbol{P}_0 = \boldsymbol{Y}_1 \boldsymbol{M}_1 \boldsymbol{Y}_1^t$, the regularization parameter μ and the PSF. To illustrate the shift invariance of the gain vector, we use a 1-D example in which the PSF is given in figure 2.1, the matrix $\boldsymbol{P}$ is taken to be the identity matrix, and the parameter μ is set to 0.05. These choices yield the series of gain vectors given in figure 2.2. It is clear that the gain vectors converge. The speed of convergence is rapid as is illustrated in figure 2.3 where after only 6 iterations the gain has essentially converged to a shift invariant vector. The same behavior

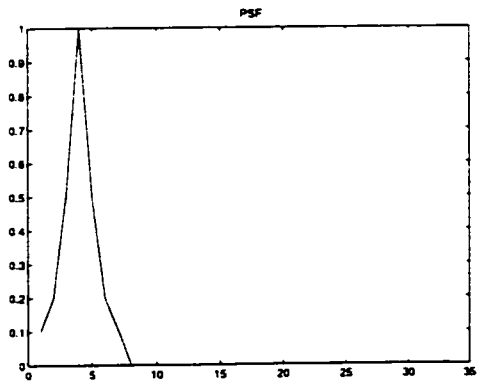


Figure 2.1: One Dimensional PSF

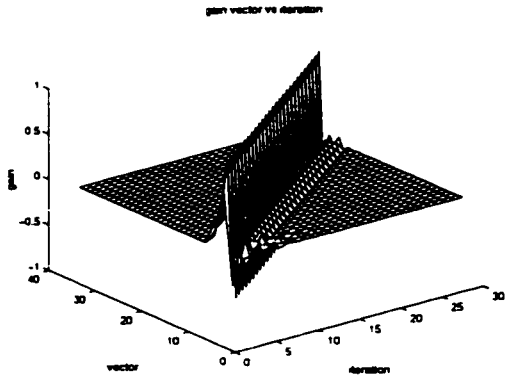


Figure 2.2: Gain vector at each iteration

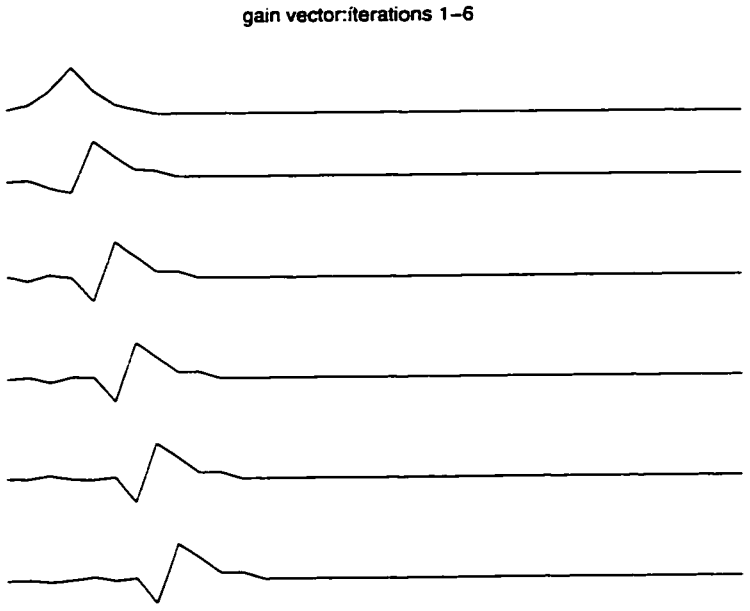


Figure 2.3: Gain vector: first 6 iterations

is observed in two dimensions. In order to reduce the computational load, the gain vector is calculated ahead of time by iterating over the Chandrasekhar equations excluding Equation 2.54 which updates the estimated image.

While the RUKF introduces sub-optimalities associated with the extension of an auto-regressive model to 2D and the *a priori* reduction on the support of the gain, no such sub-optimalities are introduced by the Chandrasekhar filter. The computational

burden associated with a large state vector is reduced by factoring the equations and then exploiting the stationarity of the gain.

2.3 Square Root Factorizations

We present here a method based on square root factorizations which attempts to improve the stability of the Chandresakhar equations. Square root implementations are used since it is observed in practice that the covariance matrices being propagated become smaller and smaller and consequently the finite precision capabilities of the computer lead to non-negligible errors which cause the algorithms to diverge. If we instead propagate the square roots (ie. $\boldsymbol{M} = \boldsymbol{S}\boldsymbol{S}^t$; $\boldsymbol{S}$ is the square root factor of $\boldsymbol{M}$), the matrices being propagated will contain larger values. Another important consequence is that if $\boldsymbol{M}$ is non-negative definite, by propagating $\boldsymbol{S}$ we guarantee that $\boldsymbol{M}$ retains this property. This is important as the loss of non-negative definiteness of this matrix is associated with the instability of the system. The developments presented in this section follows those presented in (Besnerais and Goussard 1993, Morf and Kailath 1975) for similar problems.

We restate the model and the Kalman filtering equations which will prove useful in our derivation:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k = const \tag{2.68}$$

$$\boldsymbol{y}_k = \boldsymbol{h}_k^t \boldsymbol{x}_k + \boldsymbol{v}_k \tag{2.69}$$

$$\boldsymbol{h}_{k+1} = \boldsymbol{D}\boldsymbol{h}_k \tag{2.70}$$

Kalman equations

$$\hat{\boldsymbol{x}}_{k+1|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{k}_k e_k^{-1}[\boldsymbol{y}_{k+1} - \boldsymbol{H}_k \hat{\boldsymbol{x}}_{k+1|k}] \tag{2.71}$$

$$e_k = \boldsymbol{h}_k^t \boldsymbol{P}_{k|k-1} \boldsymbol{h}_k + r_k \tag{2.72}$$

$$\boldsymbol{k}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{h}_k \tag{2.73}$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{k}_k e_k^{-1} \boldsymbol{K}_k^t \tag{2.74}$$

We begin by factoring $\delta \boldsymbol{P}_0$ as $\delta \boldsymbol{P}_0 = \boldsymbol{Y}_0 \boldsymbol{S} \boldsymbol{Y}_0^t$ where $\boldsymbol{S}$ is a signature matrix composed of 1 or -1 on the main diagonal which remains constant over during the recursions (Kailath, Vieira and Morf 1978). This is done as follows:

$$\begin{aligned}
\delta \boldsymbol{P}_1 &= \boldsymbol{P}_1 - \boldsymbol{D} \boldsymbol{P}_0 \boldsymbol{D}^t \\
&= \left[\begin{array}{c|c|c} 1 & \nu^t & 0^t \\ \hline \nu & & \\ \hline 0 & & \end{array}\right] - \boldsymbol{k}_1 e_1^{-1} \boldsymbol{k}_1^t \\
&= \left[\begin{array}{c|c|c} 1 & 0 & \frac{\boldsymbol{k}_0}{\sqrt{r_0}} \\ \hline \nu & \nu & \end{array}\right] \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & \nu^t \\ 0 & \nu^t \\ \frac{\boldsymbol{k}_0}{\sqrt{r_0}}^t & \end{bmatrix}
\end{aligned} \tag{2.75}$$

In the case when $\boldsymbol{P}$ is well structured such as Toeplitz or identity, this factorization is possible since the matrix $\delta \boldsymbol{P}_1$ has non-zero entries only in the first row and first column.

Ultimately we wish to express the square roots of quantities at instant k in terms of those at instant $k-1$. We begin by introducing the following notation:

$$\boldsymbol{A} \sim \boldsymbol{B} \quad \Leftrightarrow \boldsymbol{A} \boldsymbol{J} \boldsymbol{A}^t = \boldsymbol{B} \boldsymbol{J} \boldsymbol{B}^t \tag{2.76}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are matrices and $\boldsymbol{J}$ is a signature matrix. We note that $\boldsymbol{A} \sim \boldsymbol{B}$ implies that $\boldsymbol{A}$ and $\boldsymbol{B}$ are equivalent to within a $\boldsymbol{J}$-orthogonal transformation (Golub

and Van Loan 1989), i.e.:

$$\exists \boldsymbol{T} \text{ such that } \boldsymbol{AT} = \boldsymbol{B} \text{ and } \boldsymbol{TJT}^t = \boldsymbol{J}. \tag{2.77}$$

The above observation is the key to developing the square root factorization method. We will attempt to show that the gain $\boldsymbol{k}e^{-1}$ at time k can be related to the gain at time $k+1$ by a matrix transformation. That is we will show that if the matrix $\boldsymbol{A}$ in 2.77 is suitably chosen with the quantities at time k, we can find the quantities at time $k+1$ by a matrix transformation $\boldsymbol{T}$. We will now demonstrate that

$$\begin{bmatrix} e_{k-1}^{1/2} & \boldsymbol{H}_k\boldsymbol{Y}_{k-1} \\ \boldsymbol{DK}_{k-1}e_{k-1}^{-1/2} & \boldsymbol{Y}_{k-1} \end{bmatrix} \sim \begin{bmatrix} e_k^{1/2} & 0 \\ \boldsymbol{k}_k e_k^{-1/2} & \boldsymbol{Y}_k \end{bmatrix} \tag{2.78}$$

where the quantities are the same as those used in the Chandresakhar filter except that $\boldsymbol{Y}$ now factors $\delta\boldsymbol{P}$ as indicated in Equation 2.75. Also the matrix $\boldsymbol{J}$ is given by

$$\boldsymbol{J} = \begin{bmatrix} 1 & 0^t \\ 0 & -\boldsymbol{S} \end{bmatrix} \tag{2.79}$$

where $\boldsymbol{S}$ is the signature matrix used to obtain $\delta\boldsymbol{P}_0 = \boldsymbol{Y}_0\boldsymbol{S}\boldsymbol{Y}_0^t$. The matrices on the LHS and RHS of 2.78 are referred to as the pre-array and post-array respectively. The pre-array contains the quantities at time $k-1$ while the post-array contains the quantities at time k. It is clear that if we demonstrate such a relationship, then the square roots of the quantities at time k are related to those at time $k-1$ by a J-orthogonal transformation matrix. In particular we note that the quantities $\boldsymbol{k}$ and e appear in the post-array which will ultimately be used to calculate the gain $\boldsymbol{k}_k e_k^{-1}$. In order to prove this result, we adapt the approach used by Morf (Morf and Kailath 1975) to our case.

We first observe that

$$\begin{bmatrix} \boldsymbol{P}_{k+1}^{1/2} & 0 \end{bmatrix} \sim \begin{bmatrix} \boldsymbol{D}\boldsymbol{P}_{k}^{1/2} & \boldsymbol{Y}_k \end{bmatrix} \tag{2.80}$$

A straightforward multiplication of matrices and use of the Kalman equations yields:

$$\begin{bmatrix} r_k^{1/2} & \boldsymbol{H}_k\boldsymbol{P}_k^{1/2} \\ 0 & \boldsymbol{D}\boldsymbol{P}^{1/2} \end{bmatrix} \sim \begin{bmatrix} e_k^{1/2} & 0 \\ \boldsymbol{D}\boldsymbol{K}_k e_k^{-1/2} & \boldsymbol{D}\boldsymbol{P}_{k+1}^{1/2} \end{bmatrix} \tag{2.81}$$

We now develop by adding a block of zeros:

$$\begin{bmatrix} r_k^{1/2} & \boldsymbol{H}_k\boldsymbol{P}_k^{1/2} & | & 0 \\ 0 & \boldsymbol{P}_k^{1/2} & | & 0 \end{bmatrix} \sim \begin{bmatrix} r_k^{1/2} & \boldsymbol{H}_k\boldsymbol{D}\boldsymbol{P}_{k-1}^{1/2} & | & \boldsymbol{H}_k\boldsymbol{Y}_{k-1} \\ 0 & \boldsymbol{D}\boldsymbol{P}_{k-1}^{1/2} & | & \boldsymbol{Y}_{k-1} \end{bmatrix} \text{ by using 2.80} \tag{2.82}$$

$$\sim \begin{bmatrix} r_k^{1/2} & \boldsymbol{H}_{k-1}\boldsymbol{P}_{k-1}^{1/2} & | & \boldsymbol{H}_k\boldsymbol{Y}_{k-1} \\ 0 & \boldsymbol{D}\boldsymbol{P}_{k-1}^{1/2} & | & \boldsymbol{Y}_{k-1} \end{bmatrix} \tag{2.83}$$

$$\sim \begin{bmatrix} e_{k-1}^{1/2} & 0 & | & \boldsymbol{H}_k\boldsymbol{Y}_{k-1} \\ \boldsymbol{D}\boldsymbol{k}_{k-1}e_{k-1}^{-1/2} & \boldsymbol{D}\boldsymbol{P}_k^{1/2} & | & \boldsymbol{Y}_{k-1} \end{bmatrix} \text{ from 2.81} \tag{2.84}$$

Using the Kalman equations and 2.80 also allows us to develop the left hand side of Equation 2.81 as follows:

$$\begin{bmatrix} r_k^{1/2} & \boldsymbol{H}_k\boldsymbol{P}_k^{1/2} & | & 0 \\ 0 & \boldsymbol{P}_k^{1/2} & | & 0 \end{bmatrix} \sim \begin{bmatrix} e_k & 0 & | & 0 \\ \boldsymbol{k}_k e_k^{-1/2} & \boldsymbol{P}_{k+1}^{1/2} & | & 0 \end{bmatrix} \tag{2.85}$$

$$\sim \begin{bmatrix} e_k^{1/2} & 0 & | & 0 \\ \boldsymbol{k}_k e_k^{-1/2} & \boldsymbol{D}\boldsymbol{P}_k^{1/2} & | & \boldsymbol{Y}_k \end{bmatrix} \tag{2.86}$$

Extracting the required columns from the matrices in 2.84 and 2.86 yields the desired

result:

$$\begin{bmatrix} e_{k-1}^{1/2} & \boldsymbol{H}_k \boldsymbol{Y}_{k-1} \\ \boldsymbol{D}\boldsymbol{k}_{k-1} e_{k-1}^{-1/2} & \boldsymbol{Y}_{k-1} \end{bmatrix} \sim \begin{bmatrix} e_k^{1/2} & 0 \\ \boldsymbol{k}_k e_k^{-1/2} & \boldsymbol{Y}_k \end{bmatrix} \tag{2.87}$$

The problem now is to find the transformation matrix $\boldsymbol{T}$ such that:

$$\begin{bmatrix} e_{k-1}^{1/2} & \boldsymbol{H}_k \boldsymbol{Y}_{k-1} \\ \boldsymbol{D}\boldsymbol{K}_{k-1} e_{k-1}^{-1/2} & \boldsymbol{Y}_{k-1} \end{bmatrix} \boldsymbol{T} = \begin{bmatrix} e_k^{1/2} & 0 \\ \boldsymbol{k}_k e_k^{-1/2} & \boldsymbol{Y}_k \end{bmatrix} \tag{2.88}$$

Determining this transformation matrix is a problem which has been studied by Steinhardt and Rader in (Steinhardt 1988, Rader and Steinhardt 1986). We observe that the matrix $\boldsymbol{T}$ we seek has the property that when applied to the pre-array of Equation 2.88 it yields the first row of the post-array which is 0 everywhere except in its first entry. Two possible transformations which accomplish this task are the Householder transformation and the Givens rotation (Golub and Van Loan 1989). We will adopt the Householder transformation as studies have shown that it has desirable numerical properties (Rader and Steinhardt 1986). If we wish to perform such a transformation on an arbitrary vector $\boldsymbol{x}$, the required matrix is given by:

$$\boldsymbol{T} = I - \frac{2\boldsymbol{b}\boldsymbol{b}^t}{\boldsymbol{b}\boldsymbol{J}\boldsymbol{b}^t} \tag{2.89}$$

$$\text{where } \boldsymbol{b} = x + \sqrt{\boldsymbol{x}\boldsymbol{J}\boldsymbol{x}^t}[10\ldots0] \tag{2.90}$$

as demonstrated in (Steinhardt 1988). Here the vector $\boldsymbol{b}$ is the first row of the matrix on the LHS of Equation 2.88. We note that the transformation matrix depends on the signature matrix $\boldsymbol{J}$. Applying this transformation to the whole matrix on the LHS of 2.88 yields the matrix on the RHS. In practice, post-multiplication by the householder transformation matrix is unnecessary and computation time can be saved by using

the following procedure which fills the matrix $\boldsymbol{A}$ with the matrix $\boldsymbol{AT}$:

$$\beta = \frac{-2}{\boldsymbol{b}^t\boldsymbol{b}} \tag{2.91}$$

$$\omega = \beta \boldsymbol{A}\boldsymbol{b} \tag{2.92}$$

$$\boldsymbol{A} = \boldsymbol{A} + \omega \boldsymbol{b}^t \tag{2.93}$$

The complete algorithm is summarized in table 2.1.

Table 2.1: Square root factorization algorithm

initialize $\delta \boldsymbol{P}_0 = \boldsymbol{Y}_0 \boldsymbol{S} \boldsymbol{Y}_0^t$
for $k = 0$ to $n-1$ where n is the number of samples
$\boldsymbol{J} = \begin{bmatrix} 1 & 0 \\ 0 & -S \end{bmatrix}$
Construct the vector $\boldsymbol{b}$used in the transformation
$\boldsymbol{b} = [e_{k-1}^{1/2} \quad \boldsymbol{H}_k \boldsymbol{Y}_{k-1}]$
Construct the pre-array $\boldsymbol{A} = \begin{bmatrix} e_{k-1}^{1/2} & \boldsymbol{H}_k \boldsymbol{Y}_{k-1} \\ \boldsymbol{D}\boldsymbol{k}_{k-1} e_{k-1}^{-1/2} & \boldsymbol{Y}_{k-1} \end{bmatrix}$
now compute the post array

$$\begin{aligned} \beta &= \frac{-2}{\boldsymbol{b}^t\boldsymbol{b}} \\ \omega &= \beta \boldsymbol{A}\boldsymbol{b} \\ \boldsymbol{A} &= \boldsymbol{A} + \omega \boldsymbol{b}^t \end{aligned}$$

Compute the gain $\boldsymbol{k}_k e_k^{-1}$ from the elements in the post-array
update the estimate $\hat{\boldsymbol{x}}$ by using the newly calculated gain

$$\hat{\boldsymbol{x}}_{k+1|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{k}_k e_k^{-1} [\boldsymbol{y}_{k+1} - \boldsymbol{H}_k \hat{\boldsymbol{x}}_{k|k-1}] \tag{2.94}$$

In this chapter we have looked at recursive filtering methods based on the Kalman filter. Such filtering techniques can be used to filter images rapidly since the convergence to a shift-invariant gain can be exploited to substantially reduce computations. However the recursive filters presented in this section cannot be used when correlation is introduced in the *a priori* since the algorithms diverge. In practice, we find that even the square root factorization methods presented in this section diverge when we use a matrix $\boldsymbol{P}$ which is Toeplitz. The problem lies in the fact that the matrix $\boldsymbol{J}$ contains both 1 and -1. Consequently, we are once again faced with the problem that there is a subtraction of quantities of the same magnitude which leads to results which are positive rather than negative and vice versa. This is a consequence of the finite precision capabilities of the computer. Unfortunately it is impossible to have a matrix which has only ones on the diagonal since in the factorization of $\delta \boldsymbol{P}_1$ we must subtract the term $\boldsymbol{k}e^{-1}$.

Another important drawback of this filter is that it cannot take into account local characteristics of the image such as the presence of transitions or homogeneous zones. Attempts to filter the image adaptively lead to numerical instability which will be discussed in Section 4. Furthermore, extending the algorithm to three dimensions is impossible as it results in matrices which are much too large.

In order to deal with these drawbacks, in the article presented in the next section, we present the constrained Wiener filter which will address all the problems encountered with filters based on the Kalman filter.

Chapter 3

Article on Wiener Filtering

This chapter contains an article to be submitted to the journal IEEE Medical Imaging. The algorithms considered up to this point (Chandrasekhar and Householder) are only implementable in two dimensions due to the large matrices which result if we attempt an extension to three dimensions. Furthermore the Chandrasekhar and square root algorithms are unstable when the *a priori* covariance matrix $E[\boldsymbol{X}\boldsymbol{X}^t]$ is chosen different from identity. That is, we assume our image has some correlation.

The Wiener filter presented in this section allows us to perform filtering in three dimensions and also to introduce an *a priori* information consisting of correlation in the image to be restored.

3.1 Introduction

In this article the problem of image restoration is addressed. This problem arises in a variety of applications including satellite imaging, geophysics, and radioastronomy. Here we consider the problem of restoring a series of tomographical images. The image restoration techniques presented in this article are used in conjuction with other

image processing methods in a project which ultimately involves the construction of a personalized knee prosthesis. The development of a personalized knee prosthesis is motivated by the inadequacies of the knee prostheses currently in use. The problems with the current prostheses include: the necessity of cutting the bone to fit the shape of the prosthesis, and the difficulties involved in making revisions later since the bone must be re-cut to fit the prosthesis. In order to deal with these problems, a new type of prosthesis has been proposed which conforms to the shape of the bone. This new prosthesis is a thin personalized prosthesis which must be specifically fabricated for each individual requiring one. In order to acquire the geometry of the knee, we use X-ray tomography to obtain images of slices of the knee as this technique is non-invasive. However the blurring and noise in the images prohibits a direct extraction of contours and ultimately the surface of the knee. Consequently, we must resort to imaging techniques to process the image to obtain the required sub-millimeter precision. The image restoration phase is the first step in accomplishing this task and we will interest ourselves in this article with the development of image restoration techniques for this problem.

In order for the image restoration algorithm to be successful with respect to realistic applications in the medical field including our application, several objectives must be met. The algorithm must be able to sufficiently reduce the blur and noise present in the image. We seek a suitable compromise between reduction of the blur and suppression of the noise since these are often conflicting objectives. In particular, we seek to reduce the blur at the transitions in the image since this will lead to improved accuracy in determining the contours. By contrast, the homogeneous regions should be smoothed. We will be particularly interested in adaptive filters which update pixels locally based on whether the pixel is located in a homogeneous region or at a transition. In homogeneous regions the filter should perform a smoothing

operation while the transitions should be restored.

The algorithm must also be computationally efficient. The necessity of this is immediately seen in our case where we must restore 30 images, each roughly 200×200. Consequently, the number of pixels that must be updated is equal to 1.2 million. In addition, we require that the algorithm should be able to exploit the correlation between the slices, and consequently should be implementable in 3 dimensions. Finally, obtaining a good result from restoration algorithms often involves a judicious setting of tuning parameters. Consequently, it is desirable to have an algorithm that automatically selects the necessary parameters. In our case the parameter allows us to compromise between smoothness in the image and sharpness of transitions in the restored image.

The general image restoration problem has been addressed using several techniques including methods minimizing quadratic criteria either recursively using Kalman filtering algorithms (Demoment, Reynaud and Ségalen 1983, Woods and Radewan 1977, Woods and Ingle 1981, Demoment 1987), or iteratively using conjugate gradient based algorithms (Nagy et al. 1996, Golub and Van Loan 1989). Methods based on Markov random fields have also been developed (Geman and Geman 1984, Blake 1989, Blake and Zisserman 1987, Jeng and Woods 1990, Jeng and Woods 1991). (Banham and Katsaggelos 1997, Demoment 1989) contain general surveys of the algorithms currently being used for the general image restoration problem. Recently, Rathee proposed techniques specifically addressing the image restoration problem as it applies to CT images (Rathee et al. 1992b, Rathee, Zoly and Overton 1992c, Rathee, Zoly and Overton 1992a). Rathee proposes methods which restore the projection data and methods which restore the reconstructed images. The methods proposed in Rathee's articles all minimize quadratic criteria.

Algorithms based on the Kalman filter and the conjugate gradient which mini-

mize quadratic criteria present problems with the computational complexity when we attempt to extend the algorithms to three dimensions. Methods based on Markov random fields can be extended to three dimensions, but they are also computationally expensive. Furthermore, no parameter estimation is possible at present. The methods proposed by Rathee operate in the frequency domain and allow for adaptive filtering as a function of the local noise characteristic by exploiting the noise power spectrum. However, these methods are not suitable for adaptively filtering the image so as to smooth homogeneous zones while restoring transitions. This process will be described in detail in Section 3.4. The restoration algorithm developed here uses a spatially constrained Wiener filter to restore the image (Reichenbach and Park 1991, Reichenbach, Koehler and Strelow 1995). The constrained Wiener filter is calculated and the restoration is accomplished by convolution. This presents several advantages over other methods. In particular, the constraining of the support leads to a filtering operation which is much less costly than restoration based on the Kalman filter, the conjugate gradient or Markov Random Fields. Furthermore, the computational complexity remains manageable when the filter is extended to three dimensions. This is not true of the Kalman filter or conjugate gradient algorithm both of which require the multiplication matrices which become much too large in three dimensions. The filter also lends itself well to adaptive filtering in which the filter can be chosen to either smooth or restore edges as a function of the local characteristics of the image.

The contribution of this article is threefold. Firstly, after adapting the constrained Wiener filter developed by Reichenbach in (Reichenbach and Park 1991) to the problem of restoring CT images and then extending the results to three dimensions, we propose an original Cross Validation based algorithm for the estimation of the Wiener filter parameter which controls the degree of smoothing in the image. This allows the resulting algorithm to be performed in an un-supervised manner which is advanta-

geous in the medical field where the people using the algorithms will most often be unfamiliar with its details.

Secondly, we develop a method using the constrained Wiener filter to adaptively filter the images. Using the cross-validation technique we develop, we demonstrate that it is possible to filter the image adaptively in an un-supervised manner.

Thirdly, we study the effects of considering a varying point spread function (PSF) under varying conditions. We observe that accounting for the variations in the PSF does not produce an appreciable improvement unless the variation in the PSF is unrealistically large. Although Rathee (Rathee et al. 1992b) gives an algorithm for image restoration which takes into account the varying PSF, his results contain no comparison with algorithms that do not take the spatial variation into account. Our results suggest that comparable results can be obtained by using a spatially invariant PSF.

The paper is organized as follows. A mathematical formulation of the Wiener filter is given in Section 3.2 along with the development of the constrained Wiener filtering. We then present our algorithm for the estimation of parameters in Section 3.3 followed by a description of the adaptive filtering we propose in Section 3.4. We then turn to the results in Section 3.5 where we first demonstrate that there is little to be gained in taking into account the effects of a varying PSF. The results of the parameter estimation are then presented followed by the results of using an un-supervised adaptive filtering. Finally a conclusion is given in Section 3.6

3.2 Problem Formulation and Wiener Filtering

For the sake of clarity we first consider the problem in two dimensions and then indicate how to extend the algorithm to three dimensions. Our approach is similar to

that presented in (Reichenbach et al. 1995). We model our system by the equation

$$y(i,j) = \sum_k \sum_l h(k,l)x(i-k, j-l) + n(i,j) \tag{3.1}$$

In words, the observed image y is obtained by a convolution between the point spread function h and the image being sought x, and addition of noise n. This model has been used successfully in the context of image restoration. Rathee (Rathee et al. 1992c) demonstrates the reasons as to why this is a reasonable approximation. We assume here that the point spread function is invariant over the image. It is shown in Section 3.5.1 that this is a reasonable approximation as accounting for the variation in the PSF leads to little or no improvement. Equation 3.1 can be rewritten in matrix form as follows:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \ . \tag{3.2}$$

Here $\boldsymbol{y}$ is the observed image, $\boldsymbol{n}$ is the noise, and $\boldsymbol{x}$ is a vector containing the samples of the image to be estimated inserted row by row. In 2-D $\boldsymbol{H}$ is a Toeplitz block Toeplitz matrix constructed from the samples of the PSF. This matrix equation is equivalent to Equation 3.1.

The image restoration problem which consists of solving for $\boldsymbol{x}$ in Equation 3.2 is ill-posed. That is, simple solutions such as the least squares solution are unstable in the sense that a small change in the observation may lead to a large change in the estimate. Clearly this property is undesirable. One approach to deal with this phenomenon is to adopt regularization methods. In these methods, *a priori* knowledge is introduced into the system and a compromise is sought between the *a priori* knowledge and fidelity to the observations.

In order to introduce the *a priori* we adopt a Bayesian approach in which *a priori* information can be introduced in the form of probability densities. Choosing Gaussian densities for the noise and image leads to simple and computationally inexpensive solutions. Since one of our objectives is speed we will use Gaussian densities here. However using Gaussian densities leads to smoothing in the estimated images so we will develop adaptive filtering methods to compensate for this effect. We assume that the object $\boldsymbol{x}$ to be estimated is Gaussian and stationary. We will see that the assumption of stationarity is critical in the development of the basic constrained Wiener filter. As an initial approximation, the noise $\boldsymbol{n}$ is also assumed white, Gaussian, with variance r_n and independent of the object to be restored. However in practice correlation is observed in the noise originating primarily from the reconstruction algorithms used to yield the tomographic image.

In order to obtain a solution to Equation 3.2 we minimize the expected least squares criterion which leads to well-known solutions which are simple and efficient (Reichenbach and Park 1991). The expected least squares criterion is given by:

$$E[\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2] \tag{3.3}$$

It is well-known that minimizing Equation 3.3 yields

$$\boldsymbol{x} = \boldsymbol{W}\boldsymbol{y} \tag{3.4}$$

where $\boldsymbol{W}$ is expressed in terms of the correlation functions as follows:

$$\boldsymbol{w} = \boldsymbol{R}_{yy}^{-1}\boldsymbol{r}_{xy} \tag{3.5}$$

The solution given by Equation 3.4 is usually computed in the Fourrier transform

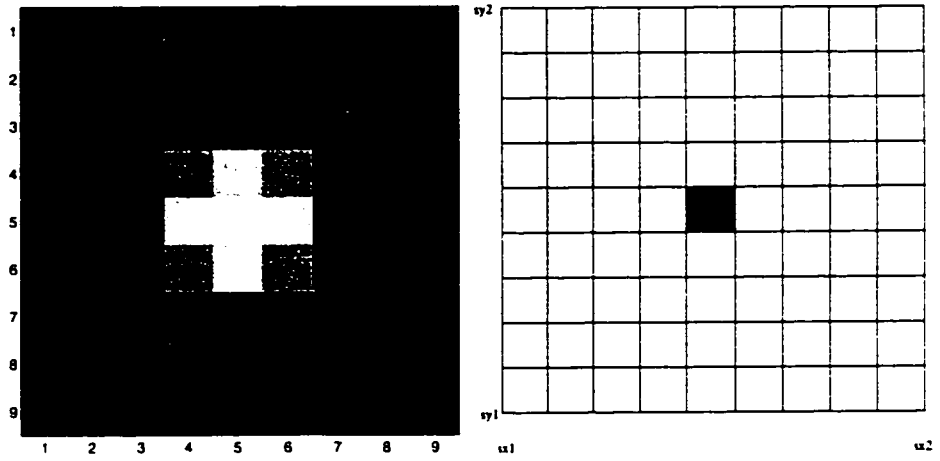
domain. The problem with this solution is that it treats the image globally and therefore we cannot take local characteristics of the image into account. Consequently we choose to adopt a local criterion as opposed to the global criterion given by Equation 3.3. The new criterion is given below:

$$E[(x_{i,j} - \hat{x}_{i,j})^2] \tag{3.6}$$

Here we choose to minimize the expected least squares error for a given pixel. We also wish to constrain the support of the Wiener filter since this will lead to a reduced computational complexity. In our problem the PSFs are no larger than 9 pixels wide. Consequently we lose very little by updating a pixel based on the information contained in neighboring pixels located within a radius equal to the size of the PSF since the significant information is contained in this region. Reichenbach (Reichenbach et al. 1995) demonstrates that constraining the filter to operate on these pixels yields filters which are only slightly sub-optimal to the unconstrained filter. The support region is illustrated in Figure 3.1. We seek a solution of the form $\hat{x}_{i,j} = \boldsymbol{w}_{i,j}^t \tilde{\boldsymbol{y}}_{i,j}$ where $\tilde{\boldsymbol{y}}_{i,j}$ is the sub-vector containing the samples in the neighborhood of the pixel $x_{i,j}$. The relevant neighborhood is determined by the width and height of the PSF as illustrated in Figure 3.1. Consequently the support of the filter $\boldsymbol{w}$ is also limited to this support given by $[s_x^1, s_x^2] \times [s_y^1, s_y^2]$. To obtain a solution we minimize Equation 3.3 for a given pixel to obtain:

$$\boldsymbol{w}_{i,j} = \boldsymbol{R}_{yy}^{-1} \boldsymbol{r}_{xy} \tag{3.7}$$

where we drop the subscripts i and j on $\boldsymbol{R}_{yy}$ and $\boldsymbol{r}_{xy}$ to simplify the notation. This is the same as Equation 3.5 except that the supports of the matrices are constrained



(a) PSF (b) support region

Figure 3.1: PSF and support region of constrained Wiener Filter

according to the constraints made on the observation $\boldsymbol{y}$ so that we have:

$$r_{xy}(k,l) = \sum_{o=sx1}^{sx2} \sum_{p=sy1}^{sy2} w(o,p) r_{yy}(k-o, l-p) \tag{3.8}$$

where r_{xy} and r_{yy} are the cross-correlation and auto-correlation functions given by the following equations:

$$r_{xy}(m,n) = \sum_{k,l} h(k,l) r_{xx}(m+k, n+l) \tag{3.9}$$

$$r_{yy}(m,n) = \sum_{i,j,k,l} h(k,l)h(i,j) \; r_{xx}(m-k+i, n-l+j) + r_n \delta(k)\delta(l) \tag{3.10}$$

where h contains the samples of the PSF and $\boldsymbol{r}_{xx}$ is the autocorrelation function of $\boldsymbol{x}$.

The *a priori* knowledge is introduced in the algorithm via the Equations 3.9 and 3.10 through the variables $r_{xx}(i,j)$ and r_n. We now exploit our hypothesis of

stationarity in the image. We observe that for a stationary image the matrix $\boldsymbol{R}_{yy}$ and $\boldsymbol{r}_{xy}$ do not depend on i and j, that is they are spatially invariant quantities. Consequently, the filter $\boldsymbol{w}$ is also spatially invariant such that we can use it to restore the object by a simple 2-D convolution between $\boldsymbol{w}$ and our observed image $\boldsymbol{y}$. We note that $\boldsymbol{r}_{xx}$ can be written as $r_x \cdot \overline{\boldsymbol{r}}_{xx}$ where $\overline{\boldsymbol{r}}_{xx}$ denotes the normalized autocorrelation function and r_x is the variance of $\boldsymbol{x}$. By factoring out r_x from Equation 3.10 in the expression for $\boldsymbol{r}_{yy}$ we find that the filter is adjusted by a parameter $\lambda = r_n/r_x$:

$$\boldsymbol{r}_{xy}(m,n) = r_x \sum_{k,l} h(k,l)\overline{r}_{xx}(m+k, n+l) \tag{3.11}$$

$$\begin{aligned} \boldsymbol{r}_{yy}(m,n) &= \sum_{i,j,k,l} h(k,l)h(i,j) \\ &\quad r_{xx}(\overline{r}_{xx}(m-k+i, n-l+j) + \frac{1}{\lambda}\delta(k)\delta(l)) \end{aligned} \tag{3.12}$$

$$\boldsymbol{w} = \lambda \overline{\boldsymbol{R}}_{yy}^{-1} \overline{\boldsymbol{r}}_{xy} \tag{3.13}$$

where the samples of $\overline{\boldsymbol{R}}_{yy}^{-1}$ and $\overline{\boldsymbol{r}}_{xy}$ are obtained from Equations 3.9 and 3.10 with r_x factored out. The parameter λ controls the degree of smoothing in the image. Increasing λ leads to smooth images while decreasing λ yields sharper edges. Estimation of this parameter will be considered in Section 3.3.

The extension of the algorithm to three dimensions is straightforward. We retain a two-dimensional PSF and introduce correlation in the third dimension. The equations stay the same, however, when we solve for $\boldsymbol{w}$ in Equation 3.7, $\boldsymbol{R}_{yy}$ which was Toeplitz block Toeplitz, now becomes Toeplitz block Toeplitz where each block is itself Toeplitz block Toeplitz.

3.3 Parameter Estimation

We now consider the problem of estimating λ for the constrained Wiener filter. Parameter estimation in the context of a quadratic criterion is a problem that has been solved using Maximum Likelihood (ML) (Fortier et al. 1993a), generalized maximum likelihood (GML) (Wahba 1985), L-curve (Brooks et al. 1993), Cross Validation (CV) and Generalized Cross Validation (GCV) (Galatsanos and Katsaggelos 1992). Maximum Likelihood involves maximizing the probability of the observation given the parameter. Mathematically this can be written as:

$$\lambda_{ML} = \arg\max_{\lambda} p(\boldsymbol{y}|\lambda) \tag{3.14}$$

This method has been studied in the context of the Kalman filter in (Fortier et al. 1993a) where it is shown that a fast computation of this criterion is possible using fast-Kalman filters. However it is unclear how to derive an algorithm which is based on the Wiener filter structure to minimize the probability. The same is true of the GML approach. The L-curve approach is less interesting since it is an empirical approach.

Consequently we pursue the CV approach first studied by Whaba in the context of spline smoothing problems (Wahba 1985). We note that in the case of image restoration GCV, which is a weighted version of CV, is equivalent to CV. This is the case since the weighting is based on the elements on the main diagonal of the matrix $\boldsymbol{H}$. In the image restoration problem with a spatially invariant PSF, the elements along this diagonal are constant so the resulting weights are identical.

CV techniques were adapted to the problem of estimating the parameter λ in the

following well known ridge regression estimate for $\boldsymbol{x}$ in Equation 3.2

$$\hat{\boldsymbol{x}} = (\boldsymbol{H}^t\boldsymbol{H} + \lambda E[\boldsymbol{X}^t\boldsymbol{X}])^{-1}\boldsymbol{H}^t\boldsymbol{y} \tag{3.15}$$

This problem has been extensively studied in the literature (see (Galatsanos and Katsaggelos 1992) or (Thompson et al. 1991) for a survey).

We adapt the CV approach to the case of the constrained Wiener filter. We will see that adopting this approach will yield a computationally efficient algorithm for estimating λ in the case of the constrained Wiener filter.

The CV algorithm consists of removing a sample from $\boldsymbol{y}$ and then predicting its value from the rest of the data. The main idea is that minimizing this prediction error over all samples will yield a reasonable choice of parameters. More precisely, the criterion is given by:

$$CV(\lambda) = \sum_k [y_k - \hat{y}_k^{-k}]^2 \tag{3.16}$$

where $\hat{y}_k^{-k}$ denotes the kth sample of $\boldsymbol{y}$ predicted from the image restored from all the data except the kth pixel. The sum is taken over the pixels in the observed image.

An important result obtained by Li in (Li et al. 1985, Li 1986) states that:

$$\lim_{M\to\infty} [\frac{\|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{H}\hat{\boldsymbol{x}}(\mu_{gcv}, \boldsymbol{y})\|^2}{\inf_{\mu>0}\|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{H}\hat{\boldsymbol{x}}(\mu, \boldsymbol{y})\|^2}] = 1 \tag{3.17}$$

where $\hat{\boldsymbol{x}}(\mu, \boldsymbol{y})$ is the estimate of $\boldsymbol{x}$ using an arbitrary $\mu > 0$ and M is the number of pixels. In words, we have that for an infinite number of pixels, the μ predicted by GCV minimizes the prediction error $\|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{H}\hat{\boldsymbol{x}}(\mu, \boldsymbol{y})\|^2$. Although we have a finite number of pixels this result still strongly motivates the use of GCV.

For the sake of clarity, we present the algorithm which evaluates Equation 3.16 in

one dimension and then indicate how it can be extended to higher dimensions. Our major problem lies in evaluating $\hat{y}_k^{-k}$ in 3.16. For this purpose, we must first calculate $\hat{\boldsymbol{x}}^{-k}$ which is the image restored from all the data except the kth pixel:

$$\hat{y}_k^{-k} = (\boldsymbol{H}\hat{\boldsymbol{x}}^{-k})_k = \boldsymbol{h}_k^t \hat{\boldsymbol{x}}^{-k} \tag{3.18}$$

where $\boldsymbol{h}_k^t$ denotes the kth row of $\boldsymbol{H}$ and $(\boldsymbol{H}\hat{\boldsymbol{x}}^{-k})_k$ is the kth sample of the vector matrix-vector product $(\boldsymbol{H}\hat{\boldsymbol{x}}^{-k})$. It is important to note here that the number of samples of $\boldsymbol{x}^{-k}$ required to evaluate Equation 3.18 is equal to the number of samples in the PSF. We now must consider the problem of evaluating the required samples of $\hat{\boldsymbol{x}}^{-k}$. We first note that we can write

$$\hat{\boldsymbol{x}}^{-k} = \tilde{\boldsymbol{W}}\boldsymbol{y}^{-k} \tag{3.19}$$

$$\hat{y}_k^{-k} = \boldsymbol{h}_k^t \tilde{\boldsymbol{W}}\boldsymbol{y}^{-k} \tag{3.20}$$

where $\boldsymbol{y}^{-k}$ is the observation vector $\boldsymbol{y}$ with the kth pixel set to 0. Each row of matrix $\tilde{\boldsymbol{W}}$ contains a distinct filter with a 0 in the kth column. These filters are evaluated from Equation 3.13 by adding the constraint that the restoration filter be 0 in the kth position. This corresponds to removing the kth sample from the support region in Equation 3.8 and the corresponding rows, columns and samples in $\boldsymbol{R}_{yy}$ and $\boldsymbol{r}_{xy}$. The problem is the calculation of all the filters in the matrix $\boldsymbol{W}$. This is done by exploiting the partitioned matrix inversion lemma on the matrix $\boldsymbol{R}_{yy}^{-1}$ which is computed once for the case where we do not add the constraint that the filter be 0 in the kth position.

In order to present the partitioned matrix inversion lemma, we begin by assuming without loss of generality that the element being removed from the support region of r_{xy} is the element at the bottom of the vector. If this is not the case, we need

only swap the element into the the bottom location and perform the corresponding swap in the matrix $\boldsymbol{R}_{yy}$ so that the corresponding column appears in the rightmost position. We now partition $\boldsymbol{R}_{yy}$ and its inverse as follows:

$$\boldsymbol{R}_{yy} = \begin{array}{c} \\ n-1 \\ 1 \end{array} \overset{\begin{array}{cc} n-1 & 1 \end{array}}{\left(\begin{array}{c|c} \tilde{\boldsymbol{R}}_{yy} & \boldsymbol{r}_{12} \\ \hline \boldsymbol{r}_{21} & r_{22} \end{array}\right)} \qquad \boldsymbol{R}_{yy}^{-1} = \overset{\begin{array}{cc} n-1 & 1 \end{array}}{\left(\begin{array}{c|c} \boldsymbol{B}_{11} & \boldsymbol{b}_{12} \\ \hline \boldsymbol{b}_{21} & b_{22} \end{array}\right)}$$

In both the above matrices, the elements in the rightmost column multiply the bottom element of the vector $\boldsymbol{r}_{xy}$ which is the element which corresponds to the element being removed from the support region of the Wiener filter. The partitioned matrix inversion lemma now yields:

$$\tilde{\boldsymbol{R}}_{yy}^{-1} = \boldsymbol{B}_{11} - \frac{\boldsymbol{b}_{12}\boldsymbol{b}_{21}}{b_{22}} \tag{3.21}$$

We note that the quantity b_{22} is a scalar since we only remove one element from the support each time. With the above equations, the rows of $\tilde{\boldsymbol{W}}$ are evaluated. Computation of the filters is done in $O(n^2)$ operations where n is the number of pixels in the constrained Wiener filter. We then use Equation 3.19 to calculate $\hat{\boldsymbol{x}}^{-k}$ and then Equation 3.18 to compute $\hat{y}_k^{-k}$.

In implementing the algorithm we note that the product $\boldsymbol{h}_k^t\tilde{\boldsymbol{W}}$ in Equation 3.20 does not depend on $\boldsymbol{y}^{-k}$. Consequently, in practice we calculate this product in advance. This yields a shift invariant filter which can then be convolved with $\boldsymbol{y}$ to obtain the vector $\hat{\boldsymbol{y}}^{-k}$ of samples $\hat{y}_k^{-k}$. We then evaluate the squared quadratic norm of the difference with $\boldsymbol{y}$ in Equation 3.16.

No analytic methods to minimize the criterion given in Equation 3.16 exist so the minimum must be searched by sampling the parameter space. We sample a range of

values of λ to find the value which minimizes the CV criterion on this space.

Extension of this algorithm to two or three dimensions is straightforward, the only difference being that the Toeplitz matrix $\boldsymbol{R}_{yy}$ becomes a Toeplitz block Toeplitz matrix in two dimensions and becomes a Toeplitz block Toeplitz matrix where each block is itself Toeplitz block Toeplitz in the case of three dimensions.

The CV algorithm can be summarized as follows:

1. Calculate the rows of $\tilde{\boldsymbol{W}}$ from $\boldsymbol{R}_{yy}^{-1}$ using the partitioned matrix inversion lemma.

2. Multiply the first row of $\boldsymbol{H}$ with $\tilde{\boldsymbol{W}}$.

3. Convolve the result with $\boldsymbol{y}$ to obtain $\hat{\boldsymbol{y}}^{-\boldsymbol{k}}$.

4. Take the square of the difference sample by sample to obtain the value of the CV criterion.

5. Iterate over the range of λ to find the minimum.

6. Use the selected value of λ to restore the image

3.4 Adaptive Filtering

We now turn out attention to the possibility of adaptively filtering the observed image in order to account for local characteristics. Rathee (Rathee et al. 1992b) proposes methods which adaptively filter the image based on the variance of the noise by using the noise power spectrum. We choose rather to adaptively filter the image based on the presence of a transition or a homogeneous zone.

The constrained Wiener filter is well suited to performing such filtering since the filter developed in Section 3.2 minimizes a local criterion in the spatial domain.

Consequently, it is possible to use a different filter for each pixel without encountering problems of stability. In particular, we propose interchanging between two constrained Wiener filters as a function of the local variance. One filter will be obtained by using a high value for the parameter λ in order to smooth homogeneous zones. The second filter can be chosen by using a smaller value of λ. We will discuss in the results section how these parameters can be chosen automatically by using the Cross Validation technique described in the previous section.

3.5 Results

We first study the effects of accounting for a varying PSF in our restoration. We find that accounting for spatial variations does not lead to superior results. Consequently in the following sections we assume a spatially invariant PSF. In this framework, we present our results for the parameter estimation problem. Finally we use the method of parameter estimation to develop a non-supervised adaptive filtering method.

3.5.1 Varying PSF

In (Rathee et al. 1992a) Rathee develops methods for estimating spatially variant PSFs. In order to be as accurate as possible, we choose to measure the PSFs directly by inserting a phantom composed of thin wires into the CT scanner. It is observed in our measurements from a CT scanner that the PSF varies with position. The PSFs can be well approximated by Gaussian. Using such approximations, we observe that the variance of the Gaussian varies by as much as 20%. This confirms theoretical results which predict a variation in the PSF associated with the finite X-ray beam width (Verly and Bracewell 1979), aliasing (Joseph, Spital and Stockman 1980), and

projection misregistration (Kijewski and Judy 1983). The PSFs in the central region have a smaller variance which progressively gets larger as we move to the extremities. In this section we consider the impact of accounting for such variation.

In order to perform the test we first construct an image which is convolved with a varying Gaussians-shaped PSF as illustrated in Figure 3.2. PSF1 is used to convolve the pixels in the central part of the image while PSF2 and PSF3 are progressively wider PSFs and are used in the outer regions. This is an approximation to the real situation in which the PSF varies continuously. We also use the image illustrated in Figure 3.3 to ensure that each PSF must restore some boundaries.

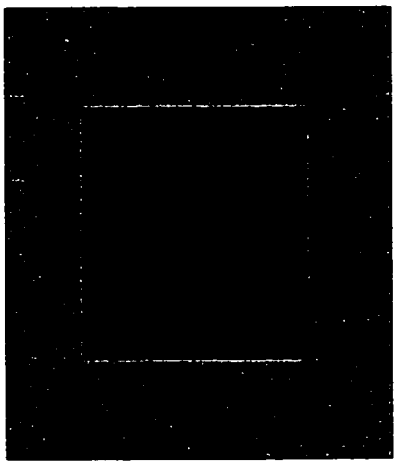


Figure 3.2: Regions for variations of the PSF

Tables 3.1, 3.2, and 3.3 illustrate the results of taking into account the variation in the PSF. The variation in the PSF listed in the table caption corresponds to the variation between PSF1 and PSF3. PSF2 corresponds to a PSF of intermediate width between PSF1 and PSF3. We consider a PSF which varies in width by 10%, 20% or 100% (*i.e.* the width doubles). An increase in width of under 20% is observed with tomographical scanners (Doré and Goussard 1997). A 100% variation is considered for the sake of comparison. In the tables, we present the results obtained by using the optimal varying filter and compare them to the results obtained using a constant

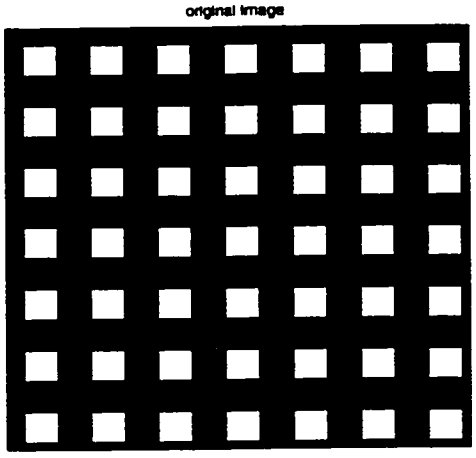


Figure 3.3: Original checkerboard image

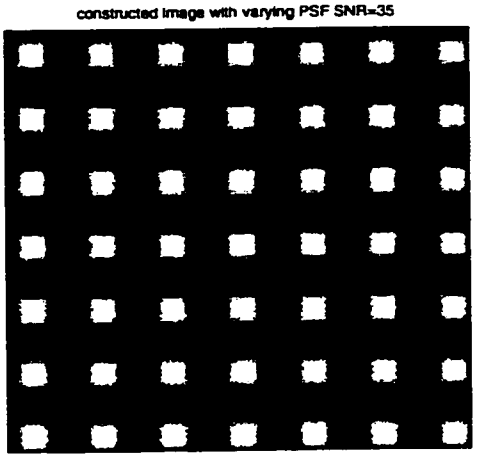


Figure 3.4: Blurred and noisy image SNR=35dB

filter associated with one of the PSFs.

The performance was measured using the average normalized square error taken pixel by pixel as well the pixel error after segmentation. We use pixel error after segmentation since we are dealing with binary images and in particular with real images we are interested in separating the two regions: bone and tissue. We use a threshold segmentation as it is simple and fast and has yielded good results. As we consider binary images, the pixel error is the number of pixels incorrectly placed in one zone or the other.

The results indicate that there is little to be gained from taking into account the varying nature of the PSF. In fact we observe that using a constant filter associated with PSF3 which is the widest PSF produces a consistently better pixel error. However this improvement is small. This can be attributed to the fact that in general, the Wiener filter leaves some blur at the edges after restoration at low SNRs. Consequently, one might expect that using a wider PSF yields a slightly better result.

Table 3.1: PSF variation 10%

	SNR	15	25	35
varying filter	normalized square error	0.1797	0.1553	0.1483
	pixel error after segmentation	372	249	203
constant filter PSF1	normalized square error	0.1708	0.1515	0.1486
	pixel error after segmentation	534	408	334
constant filter PSF2	normalized square error	0.1759	0.1533	0.1478
	pixel error after segmentation	413	284	202
constant filter PSF3	normalized square error	0.1827	0.1576	0.1501
	pixel error after segmentation	352	232	196

Table 3.2: PSF variation 20%

	SNR	15	25	35
varying filter	normalized square error	0.1747	0.1543	0.1493
	pixel error after segmentation	474	301	238
constant filter PSF1	normalized square error	0.658	0.1605	0.1661
	pixel error after segmentation	911	917	813
constant filter PSF2	normalized square error	0.6996	0.1538	0.1523
	pixel error after segmentation	596	455	414
constant filter PSF3	normalized square error	0.1811	0.1593	0.1532
	pixel error after segmentation	410	264	225

The results from this section indicate that there is little to be gained by accounting for a spatially varying PSF. In particular, in the case of tomographical imaging the

Table 3.3: PSF variation 100%

	SNR	15	25	35
varying filter	normalized square error	0.1643	0.1508	0.1524
	pixel error after segmentation	698	409	363
constant filter PSF1	normalized square error	0.5016	0.6679	0.7741
	pixel error after segmentation	1925	1970	2297
constant filter PSF2	normalized square error	0.2173	0.2359	0.2563
	pixel error after segmentation	1205	1138	954
constant filter PSF3	normalized square error	0.1888	0.1691	0.1665
	pixel error after segmentation	601	335	326

variation has been shown to be under 20% (Doré and Goussard 1997) so that we are well below the range in which some improvements are noted. In what follows, we safely assume that the PSF is spatially invariant.

3.5.2 Results of Parameter Estimation

The restoration and parameter estimation were tested on real and simulated images. For the simulations, the original object was chosen as a circle and an "H" over a constant background as illustrated in Figure 3.5. This object was chosen as it contains both rectilinear and curved contours with sharp boundaries. This allows us to measure the ability of the filter to restore sharp edges. The object was blurred by an invariant Gaussian PSF 7 pixels wide at SNRs of 5, 15, and 25dB An example at 5dB is given in Figure 3.5. The noise was Gaussian and uncorrelated. Since the noise is independent of the object, all our hypotheses hold true except that we use a representative test image rather than a realization of a Gaussian random field for our observations.

Table 3.4: Parameter estimation with varying SNR

SNR	5	15	25
real	70.1	7.01	0.701
CV chosen λ	49.2	4.64	0.134

Table 3.4 contains the results of the estimation of parameters on simulated images for various SNRs. The estimated parameters consistently underestimate the true parameters which confirms the results stated in (Fortier et al. 1993b) for the related Kalman filtering problem. However at low SNRs (15dB or less) the estimated value gives a suitable choice. At higher SNRs using the estimated parameter leads to undersmoothing. Simulations have shown that the global minimum always exists even though no proof exists to our knowledge.

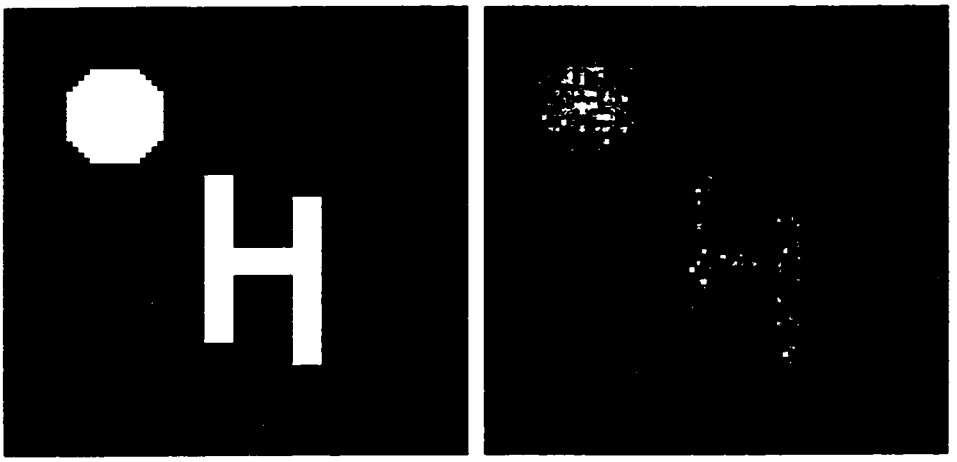


Figure 3.5: Original image and blurred image at SNR=5dB

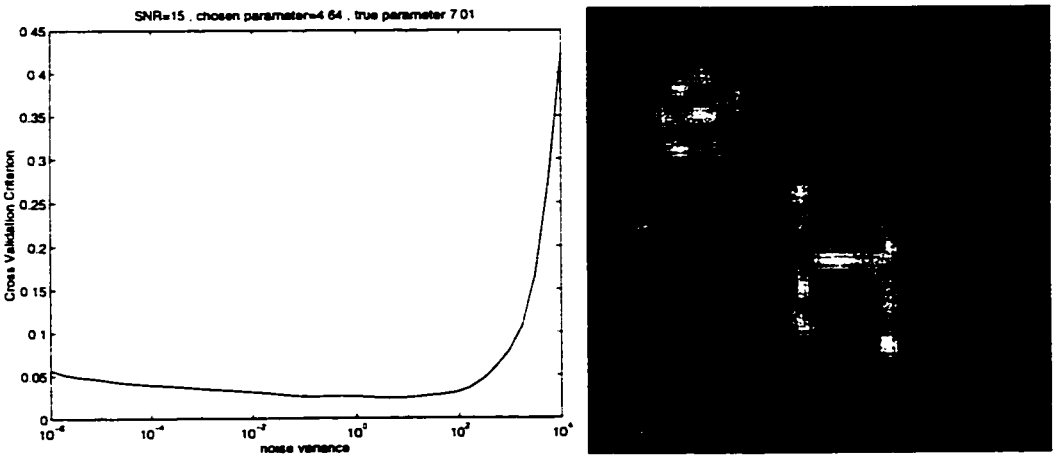
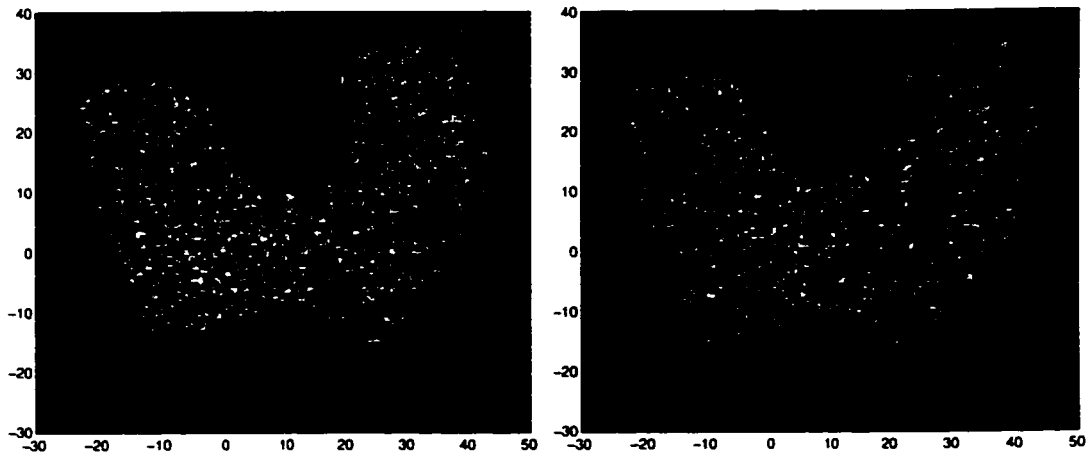


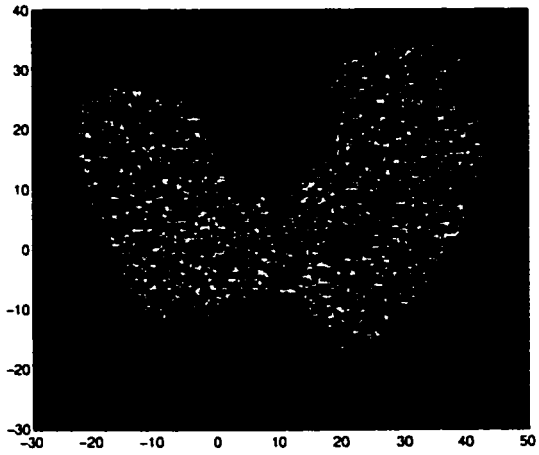
Figure 3.6: Cross Validation curve and restored image

In Figure 3.6 we present the CV curve obtained by scanning through various values of parameter λ. The minimum was then used to restore the image in 3-D assuming that the images above and below were identical with different realizations of the noise. This type of test image is used since we find that on our real images the variation between adjacent slices is small. The restoration result is shown in the right panel of Figure 3.6. We see that by using the value chosen by CV the noise has been suppressed though the edges are blurred. We will demonstrate in Section 3.5.3 how adaptive filtering can be used to reduce the blurring effect while still suppressing the noise.



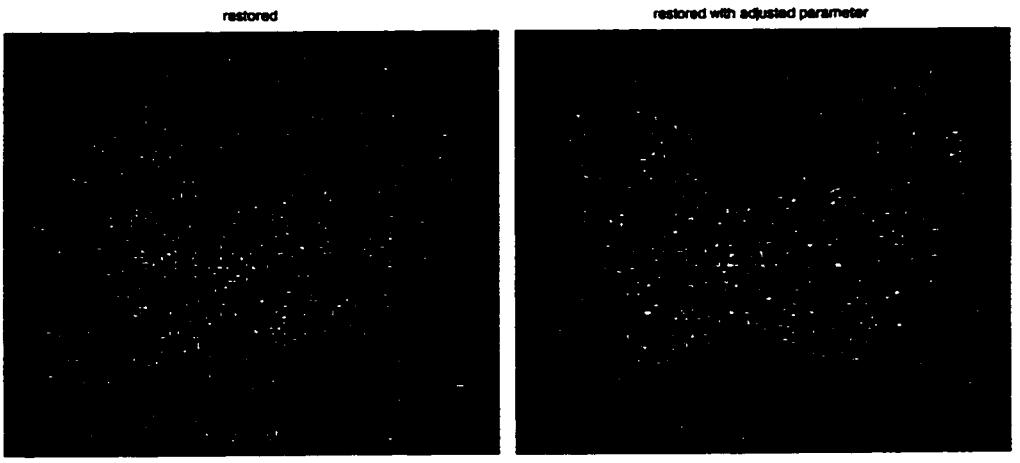
(a) lower slice

(b) middle slice



(c) upper slice

Figure 3.7: Knee slices used in the 3D restoration obtained from a CT-scanner



(a) Restored with λ_{CV} (b) Restored with $\lambda_{CV} \times 10$

Figure 3.8: Image restored with $\lambda = \lambda_{CV}$ and $\lambda = \lambda_{CV} \times 10$

Real images were obtained from a tomographic scanner on which the PSF was measured experimentally. A spatial variation of about 10% is observed in the PSF but this is neglected and the PSF associated with the central point is used. We also note that the noise in the image is correlated which goes against our hypotheses. Figure 3.7 contains real tomographic images of the three consecutive slices of the knee. The image restored by the 3D constrained Wiener filter after the selection of parameters by cross validation appears in Figure 3.8a. Since the true image is unknown, a quantitative evaluation is impossible. Qualitatively, we note that there has been an amplification of the noise. This is consistent with the fact that there is undersmoothing at higher SNRs when the parameter is chosen from the CV curve. Since the CV algorithm consistently underestimates the true noise variance, we propose adjusting the selected parameter by an empirically chosen factor of about 10 for high SNRs. The results with such an adjusted parameter are illustrated in Figure 3.8b. Once again, we qualitatively note that the blur around the edges has been reduced. Although noise is still present in the image, the level is significantly less than the

noise level in the image restored with λ_{CV}. We will see in the next section how this noise can be smoothed by using adaptive filtering.

3.5.3 Results of Adaptive Filtering

As was discussed in Section 3.2, the main advantage of the spatially constrained Wiener filter lies in its ability to perform adaptive filtering based on the local characteristics of the image. Here we are interested in smoothing the homogeneous regions of the image while restoring the transitions. Determining which zones are homogeneous and which contain transitions will be done by calculating a local variance. The parameter λ of the filters can then be determined by using CV as we will see, so that the whole process can be performed in an un-supervised manner.

In the homogeneous regions we require a Wiener filter which smoothes. Such a filter is obtained by using a high value of λ. We heuristically choose this value to be 100 times the value of λ obtained from the cross validation curve. By contrast, using a relatively low value of λ leads to images with less noise suppression, but better restoration of the transitions. This second parameter can also be chosen automatically by using a value of 2 times the value obtained from the cross validation curve. From Table 3.4 we note that at SNRs below 20 dB the value of λ_{CV} will be quite close to the actual noise variance. At SNRs above 25dB, the noise variance is under-estimated by λ_{CV} and some undersmoothing can be expected. However, we are willing to accept some undersmoothing in transition regions in exchange for sharper transitions. We will see that, in our experiments, these heuristic choices for the parameters lead to good results.

To illustrate how the adaptive filter is used in practice, we first consider the use of the local variance to separate the zones in which the two filters will be used. For each

pixel in the image, a variance on a five by five square is calculated. We then apply a threshold segmentation to yield two regions, one for each filter. If this is done on the image in Figure 3.10a, we obtain the image in Figure 3.10b where it is clear that we have been able to obtain a rough idea of the location of the edges.

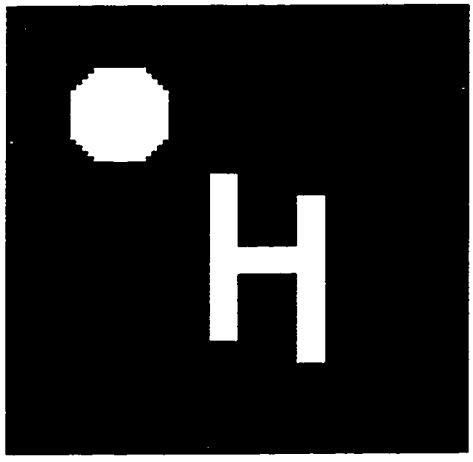
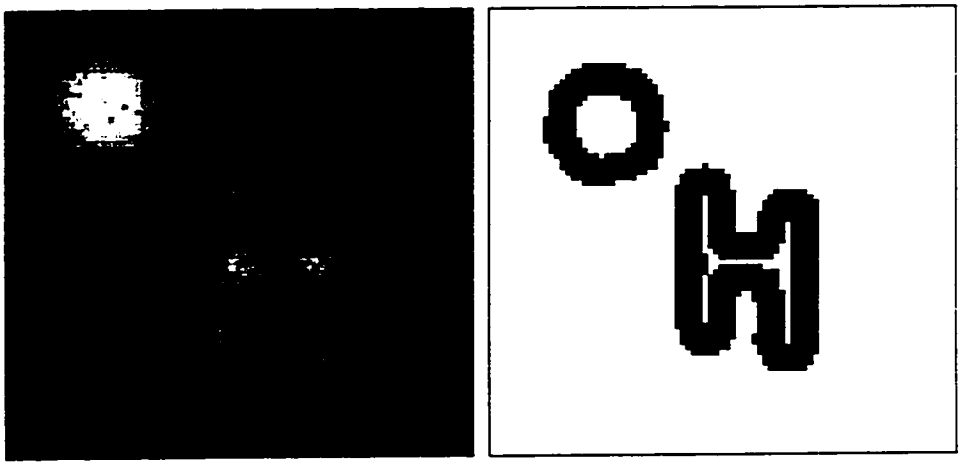


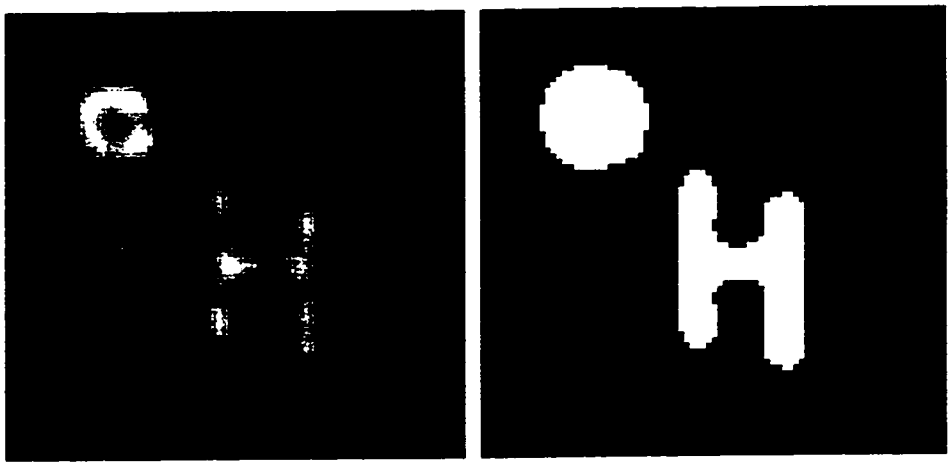
Figure 3.9: Original simulated image



(a) Blurred image; SNR=15dB (b) Local variance

Figure 3.10: Blurred image with SNR=15dB and local variance after threshold segmentation

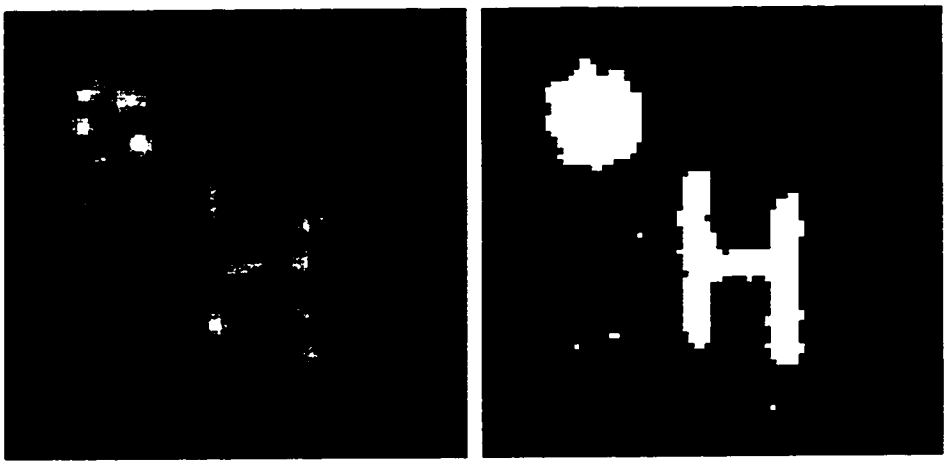
Examples of the type of results we obtain appear in Figures 3.11 to 3.13 for



(a) Restored (b) Segmented

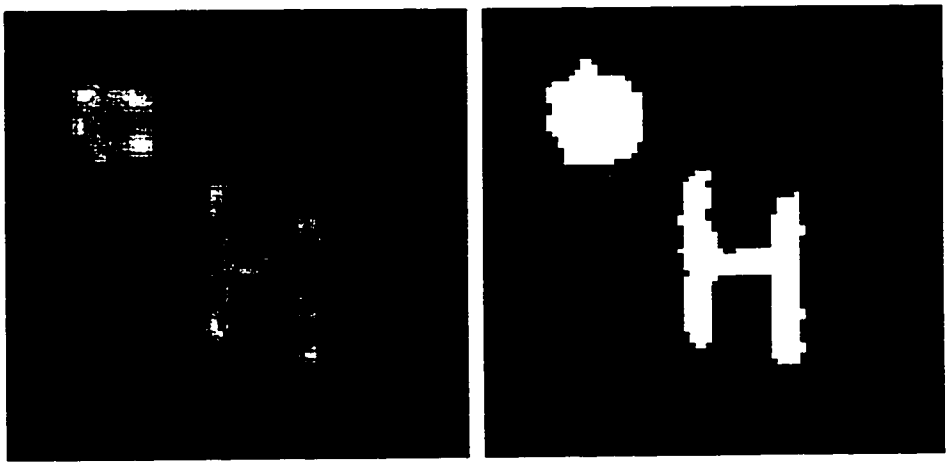
Figure 3.11: Image restored with $\lambda = 100 \times \lambda_{CV}$ then segmented

a SNR of 15dB. We include both the restored image and the image segmented by a threshold. We see in Figure 3.11 that the smoothing filter yields a segmented image which appears to be of good quality, but a close comparison of Figure 3.9 and Figure 3.11 reveals that the image obtained after restoration and segmentation contains an "O" and an "H" which are thicker then the originals by roughly one pixel along the boundary. This results from the smoothing of the edges associated with a high parameter λ. By contrast, the image restored with a low value of λ and then segmented as illustrated in Figure 3.12 contains some errant pixels outside the "O" and "H". The "O" and the "H" are better restored though not perfectly. In Figure 3.13, we see the results of the adaptive filter. After segmentation, it is clear that the errant pixels away from the "O" and the "H" have been removed. The "O" and the "H" remain roughly the same since at the transition points, the same restoring filter was used as in Figure 3.12. The numerical values for the pixel errors as well as the mean square error are given in Table 3.5 for SNRs of 15dB, 25dB and 35dB.



(a) Restored (b) Segmented

Figure 3.12: Image restored with $\lambda = 2 \times \lambda_{CV}$ then segmented



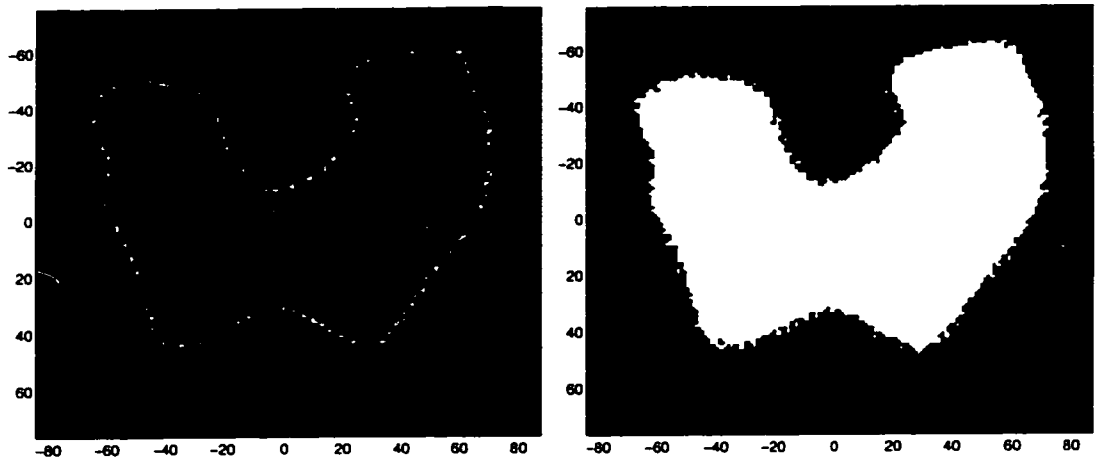
(a) Restored (b) Segmented

Figure 3.13: Image adaptively restored then segmented

Table 3.5: Comparison of Adaptive and Non-Adaptive Filtering

SNR	15	25	35
adaptive filtering pixel error	8	3	0
non-adaptive,high λ, pixel error	20	11	9
non-adaptive,low λ, pixel error	16	4	0
adaptive, square dist.	0.0589	0.0553	0.0516
non-adaptive, high λ, square dist	0.0843	0.0826	0.0824
non-adaptive, low λ, square dist	0.0653	0.0616	0.0533

We observe that the adaptive filtering method consistently outperforms either of two non-adaptive filters. This is seen in that the values for the normalized square distance between the pixels of the restored and original images are consistently lower for the adaptive filter. When we evaluate the performance after a threshold of 0.5 is used to separate the image into 2 levels we see that the adaptive filter outperforms the non-adaptive filter with a low λ at an SNR of 15dB. However, at 25dB the improvement is just one pixel and at 35dB, both yield a perfect result. This is as we expect, since the adaptive filter can be used effectively to smooth noisy regions which are homogeneous while restoring transitions. At higher SNRs, less noise is present and consequently, the need for smoothing diminishes.



(a) Restored (b) Segmented

Figure 3.14: Real Image restored by 3D Wiener filter then segmented

When we adaptively restore the image of the knee in 3D using the slices above and below illustrated in Figure 3.7, we obtain the image in Figure 3.14a. This result is clearly superior to the non-adaptively filtered image presented in Figure 3.8b. Unlike the image in Figure 3.8b, here we have been able to smooth homogeneous zones while restoring edges. The image after segmentation appears in Figure 3.14b. Once again, since the true image is not available, a quantitative evaluation is impossible. Qualitatively the result is good though we do observe some roughness along the edges.

3.6 Conclusion

In this article we have extended the constrained Wiener filter originally introduced by Reichenbach (Reichenbach and Park 1991) to 3-D for the purpose of restoring a series of tomographic images. Our main contribution lies in the development of a cross validation based algorithm for estimating the parameters in a reasonable amount of time. The algorithm underestimates the parameter at higher SNRs which leads to undersmoothing. At low SNRs the choice is accurate enough to be used in the restoration.

We have also proposed a method for adaptively filtering image using two constrained Wiener filters. The filters are adapted to the local variance in the image so as to interchange between a filter which restores edges effectively and one that smoothes noise. The parameter λ for both filters is chosen from the CV curve so that the restoration is carried out in an un-supervised manner. Results indicate that a marked improvement is obtained when using a filter adapted to the local variance as opposed to either a smoothing filter or a restoring filter.

Finally, we also consider adaptive filtering with respect to a varying PSF. However our study indicates that accounting for a varying PSF does not lead to an improvement in the restored image as measured by the pixel error after segmentation. As it is this criterion that interests us for binary images we conclude that using such adaptive filters is unnecessary.

The constrained Wiener filter provides an effective way of solving the image restoration problem. In particular, the filter is computationally inexpensive which allows us to process the large number of pixels present in a series of tomographical images. Exploiting the possibility of adaptivity leads to successfully restored edges with smoothing of the noise in homogeneous regions. Furthermore, the parameter

estimation algorithm proposed, allows this adaptive filtering to take place in an unsupervised manner.

Chapter 4

Results

In this section we compare the performances of the Chandrasekhar and Householder algorithms under varying conditions. We also present comparisons with the Wiener filter, even though some of these results already appeared in the previous section.

We first look at test images and then consider the performance of the algorithms on actual data obtained from a tomographical scanner.

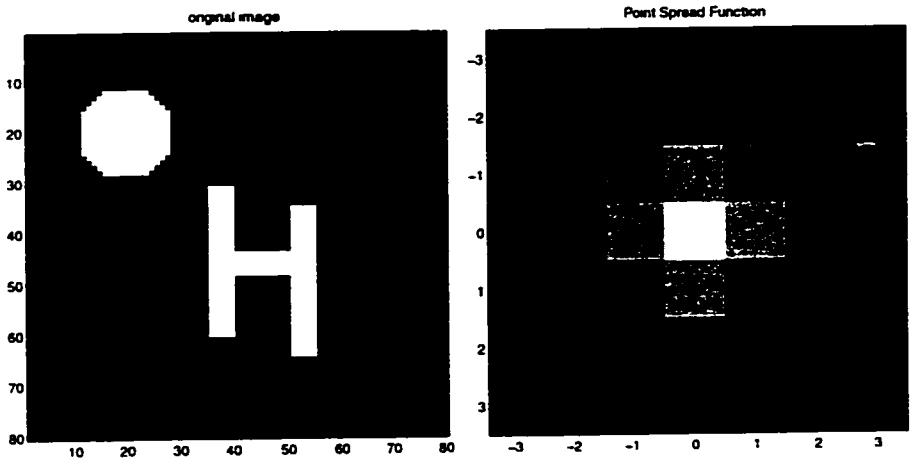
4.1 Methodology

The same method was used to test the Chandrasekhar and Householder algorithms as was used to test the Wiener filter as documented in Section 3. We note that the standard Kalman filter was not tested and is not used directly in practice since the volume of calculations is much too high. The Chandrasekhar filter is a factorization of the Kalman filter and as such yields exactly the same solution. We recall also that the Householder algorithm is also a factorization whose aim is to improve the stability of the Kalman equations when the *a priori* matrix P_0 is chosen to be Toeplitz. Therefore,

to within numerical errors, the standard Chandrasekhar filter and the square root algorithm provide the same results.

The test object is chosen as a circle and an "H" over a constant background as illustrated in Figure 4.1. This object is chosen as it presents both rectilinear and curved contours with sharp transitions. This allows us to measure the ability of the filter to restore sharp edges. This is important since in practice we are primarily interested in the ability of the method to obtain a precise knowledge of the location of the edges.

The object is blurred by an invariant Gaussian PSF 7 pixels wide at various SNRs. The PSF is also illustrated in Figure 4.1. The noise is Gaussian and white. Since the noise is independent of the object, all our hypotheses hold true except that we use a representative object with both curved and rectilinear contours with sharp transitions rather than a realization of a Gaussian random field.



(a) Original Image (b) Point Spread Function

Figure 4.1: Original image and Point Spread Function

In what follows, we examine the performance of the algorithms over a range of SNRs from 5dB to 40dB. Examining this relatively wide range is important since

the reconstruction algorithms used to generate images from raw data are tuned by parameters that allow us to compromise between the level of noise in the image and the smoothness. Most often these parameters are selected such that we find that the images obtained after reconstruction have an SNR of roughly 20dB-25dB. However, occasionally images with relatively low or high SNRs are encountered.

Three criteria will be used to judge the images. Firstly, we will use the standard mean square error since the type of filters we are using minimize quadratic criteria. We will also look at the pixel error after a segmentation which yields a binary image. The pixel error is the number of pixels falsely classified. In our simulations we use images of constant dimension so that we use this absolute error directly. This error is important since we are seeking to separate two regions: bone and tissue. We use a threshold segmentation for this purpose because it is simple, efficient and has yielded good results. Finally, we will have to resort to a qualitative evaluation of our results on the real images. This is necessary when the real images are not available since in this case no quantitative comparison is possible.

The results are grouped into three main sections as follows. In the first section we examine the results of the Chandrasekhar optimal and sub-optimal filters on simulated images. First we demonstrate the influence of the parameter μ and in particular we show that an optimal value, as measured by the mean square error, exists. We then turn our attention to a comparison of the optimal and sub-optimal Chandrasekhar filters when the matrix $\boldsymbol{P}$ is chosen to be the identity matrix. We find that the sub-optimal filter performs comparably so that subsequently we retain only this filter in the tests. We then look at the problems of stability encountered when we attempt to set the matrix $\boldsymbol{P}$ to be a Toeplitz matrix. In the same section we also examine the stability problems encountered when we attempt to use the sub-optimal filter adaptively.

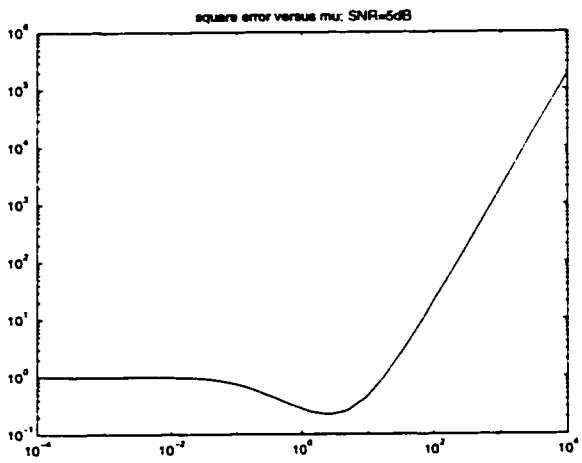
In the second section, we continue to use simulated images. In this section we compare the performance of the Chandrasekhar filter with the performance of the Wiener filter. First we compare the Chandrasekhar filter with the non-adaptive 2D and 3D Wiener filters. Subsequently we compare the Chandrasekhar filter with the adaptive 2D and 3D Wiener filters.

In the last section we turn our attention to results obtained with real images. Since with these images a quantitative comparison is not possible, we qualitatively compare the results obtained by the Chandrasekhar filter with those obtained by the adaptive 2D and 3D Wiener filters.

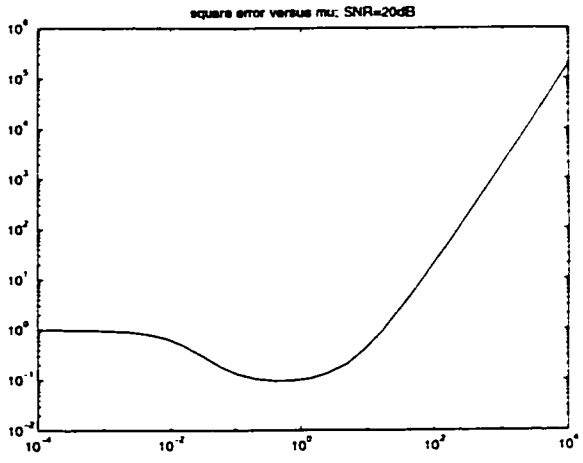
4.2 Results with simulated images

4.2.1 Influence of the parameter μ

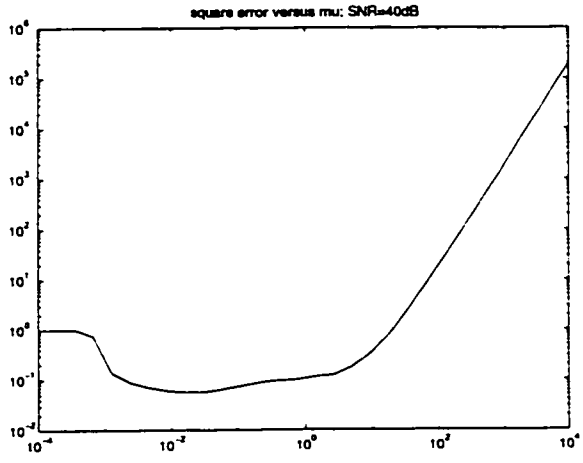
In this section we show how the setting of the parameter μ influences the results of our tests. For a given value of the SNR, we can consider the variation in the performance of the algorithm with varying μ. Such an analysis was performed and the results for SNRs of 5dB, 20dB, and 40dB are given in Figure 4.2. From this figure it is clear that an optimal value for the parameter μ exists. Methods for finding this parameter in the context of the Chandrasekhar equations are discussed in the literature survey in the section on parameter estimation. The problem with these algorithms is that they are not sufficiently accurate to be used, and some adjustments must be made in practice. Furthermore, the algorithms cannot be used in 3D because of the volume of computations associated with the updating of large matrices.



(a) 5dB



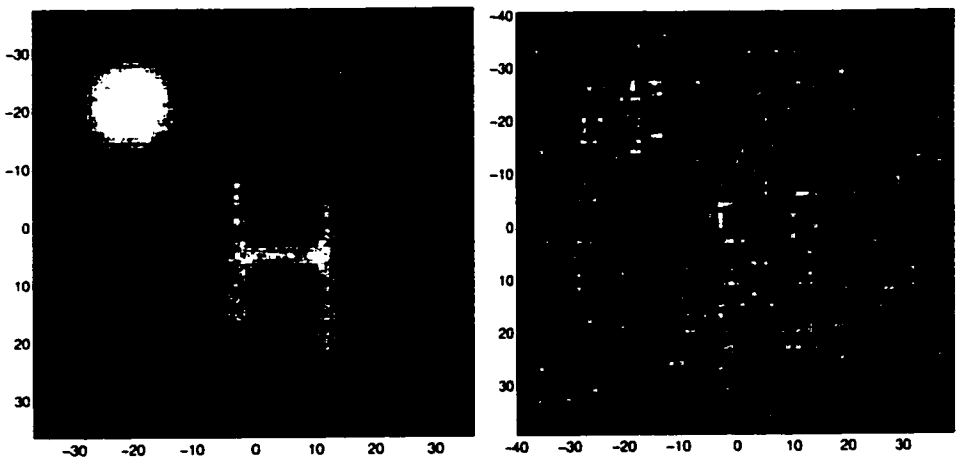
(b) 20dB



(c) 40dB

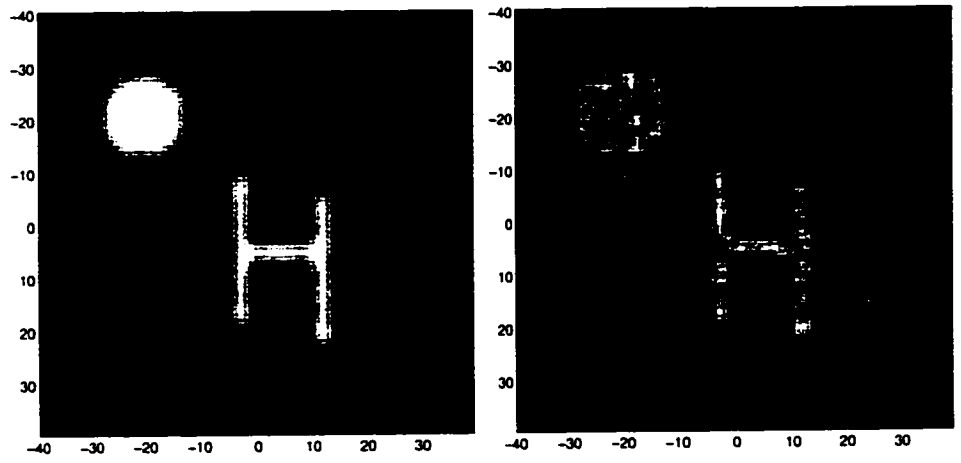
Figure 4.2: Mean square error versus μ

As illustrated in Figure 4.3 for a SNR of 20dB, we observe that as the parameter is raised we obtain smoother images. When the parameter is lowered, transitions are sharper, but we may encounter amplification of the noise. Consequently, we must accept a compromise. In subsequent sections this parameter is chosen heuristically so that we obtain good results in which an adequate compromise between suppression of the noise and restoration of the edges is obtained. We will see further that adaptive filtering in the case of the Chandrasekhar filter is impossible due to problems with stability. Consequently, we must content ourselves with accepting this compromise. By contrast, we have seen with the Wiener filter in Section 3, that un-supervised adaptive filtering is possible thanks to the Cross Validation algorithm developed in the context of Wiener filtering and the non-recursive structure of the filter.



(a) blurred image at 20dB

(b) $\mu = 0.005$



(c) $\mu = 5$

(d) $\mu = 0.08$

Figure 4.3: Restoration of image at 20dB with various values for μ

4.2.2 Results with recursive filters

For the Chandrasekhar filter, we first consider the case where we initialize the covariance matrix $\boldsymbol{P}_0$ to be the identity matrix. In other words there is no correlation in the model. We first compare the sub-optimal and optimal filters. We will see that they yield almost identical results. Consequently, we subsequently only use the sub-optimal filter and evaluate its performance before and after segmentation.

The results of the restoration process as measured by the mean square distance are given in Figure 4.4 for both the optimal and sub-optimal filters. We recall from Section 2.2.2 that the sub-optimal Chandrasekhar filter is the filter we obtain when we calculate the shift-invariant gain ahead of time and use it to restore the image. Two points are to be noted from Figure 4.4. Firstly, the performance of the filter

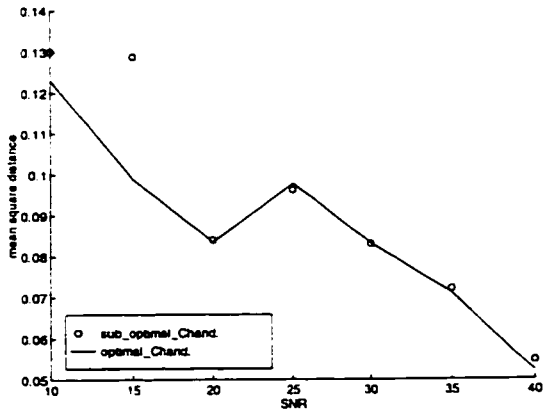
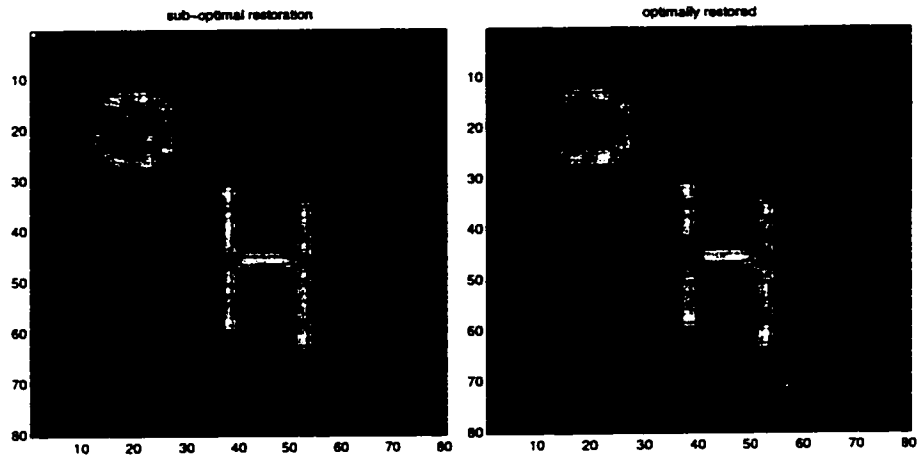


Figure 4.4: Optimal and Sub-optimal Filter Performance

falls off at lower SNRs. This is as we expect since as the noise level is increased, our observations become less accurate, and consequently the restoration becomes more difficult. From the graph it is clear that both methods yield errors which are almost identical and that there is little to be gained from using the optimal filter. At low SNRs the optimal filter performs slightly better, but this effect is slightly exaggerated

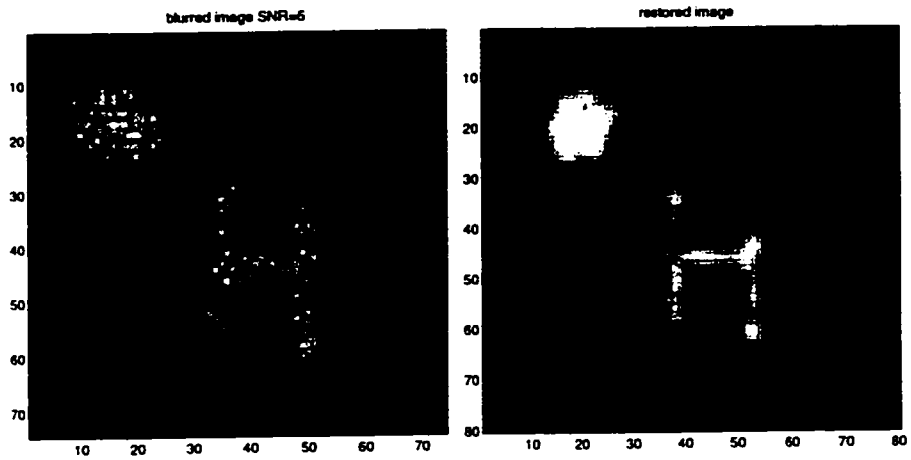
in the graph due to the scale of the y-axis which must be noted. To illustrate the



(a) Sub-optimal Restoration (b) Optimal Restoration

Figure 4.5: Comparison of Optimal and Sub-optimal restorations at 5dB using Chandrasekhar filter

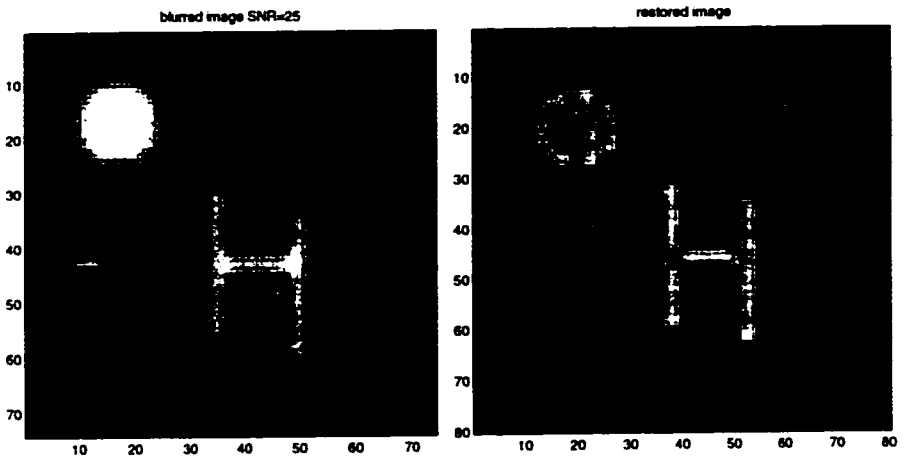
result of the sub-optimal filter with respect to the optimal filter, an example at 25dB is provided in Figure 4.5a. For comparison, the image restored with the optimal filter (that is the filter in which the gain is recalculated at every recursion) is given in Figure 4.5b. It is clear that visually the images are almost identical which is confirmed by the numerical results contained in Figure 4.4. Since the computation time is reduced by a factor of roughly 10 when using the sub-optimal filter, in practice we use the sub-optimal filter for restoration.



(a) Observed Image (b) Restored Image

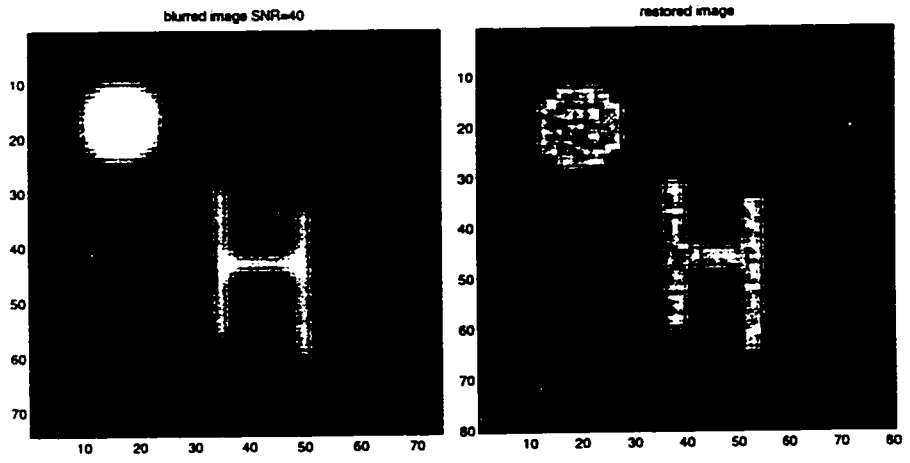
Figure 4.6: Restoration at 5dB using sub-optimal Chandrasekhar filter

The results of the sub-optimal restorations are illustrated in Figures 4.6, 4.7, and 4.8 at SNRs of 5db , 25dB and 40dB respectively. In each case the observed image is on the left and the image restored with the sub-optimal filter appears on the right. We note that at a SNR of 40dB, the restoration is qualitatively perfect since we have recovered the sharp edges of the original image. By contrast at 25dB, we notice that some of the blur still remains at the transition points. At a SNR of 5dB we have been completely unsuccessful in removing the blur. This is due to the fact that as we lower the SNR, we must raise the parameter μ to adequately suppress the noise. However this leads to smoothing of the transitions as well since as μ is increased we favor the *a priori* which is a constant image. In each case, the parameter μ was determined heuristically so as to attempt to optimize the compromise between suppression of the noise and restoration of the edges.



(a) Observed Image (b) Restored Image

Figure 4.7: Restoration at 25dB using sub-optimal Chandrasekhar filter



(a) Observed Image (b) Restored Image

Figure 4.8: Restoration at 40dB using sub-optimal Chandrasekhar filter

The pixel error at various SNRs after segmentation is given in Figure 4.9. As we expect, as the noise increases, the pixel error increases. At SNRs below 15dB, the results deteriorate rapidly, and in fact below 15dB the segmented result is insufficient for our purposes as the pixel error is too great. This results from the fact that at low SNRs the parameter μ must be increased to favor the *a priori* information which in this case is a smooth image. Consequently the resulting image is overly smooth and a large error is seen at the edges. At higher SNRs, the parameter μ can be lowered and the sharp edges are more effectively recovered resulting in a smaller pixel error after segmentation. Visually, after segmentation of the restored 10dB image by using

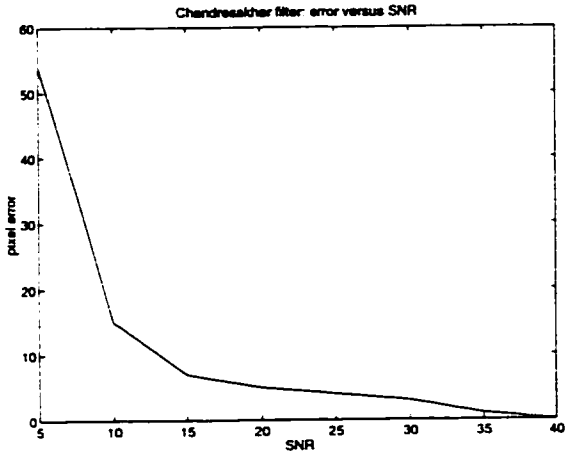


Figure 4.9: Pixel error at various SNRs

a threshold value, we obtain the image in Figure 4.10a. This can be compared to the original image by taking the absolute value of the difference pixel by pixel. This yields the image in Figure 4.10b. We can see clearly that several pixels have been falsely classified around the edges. When the SNR is lowered to 5dB we obtain the results in Figure 4.11. Here the error is consistently 1 pixel around the edges and there are also errant pixels outside the circle and "H" associated with noise that has not been adequately smoothed. By contrast the image at a SNR of 40dB yielded a perfect image after segmentation by threshold.

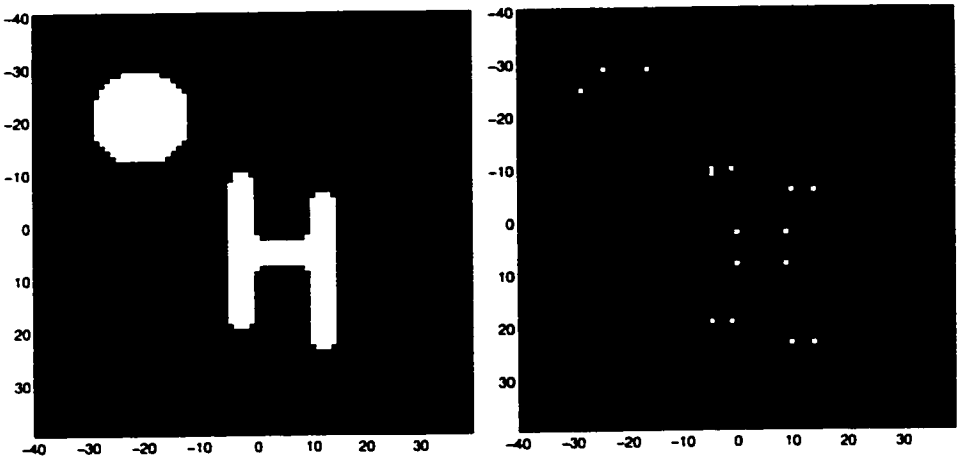


Figure 4.10: Segmented result and pixel error; SNR=10dB

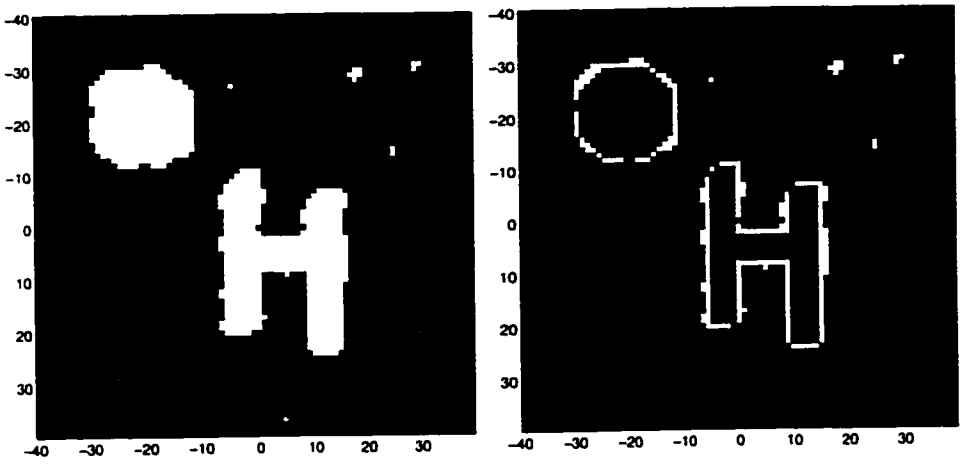


Figure 4.11: Segmented result and pixel error; SNR=5dB

4.2.3 Instability: Correlated A Priori and Adaptive Chandrasekhar Filtering

Up to this point, we have only considered filtering in which the matrix $\boldsymbol{P}_0$ is set to be the identity matrix. In other words $\boldsymbol{E}[XX^t] = \boldsymbol{I}$. However images in general are correlated at short distances. In other words nearby points resemble each other. This can be modeled by choosing the matrix $\boldsymbol{P}_0$ to be a Toeplitz-block-Toeplitz matrix and factoring in terms of $\boldsymbol{YMY}^t$ as demonstrated in Section 2.2. The equations initialized as above for the case of correlated *a priori* were programmed and tested. We find however that they are unstable. That is the gain does not converge to a stationary gain. When the algorithm is run, we observe that the quantity e_k tends to 0 so that the vector gain $\boldsymbol{k}_k e_k^{-1}$ tends to infinity.

One way to improve the stability of the algorithm is to use algorithms which use a square root factorization. One such algorithm is based on Householder transformations as discussed in Section 2.3. However, when this algorithm was tested, numerical instabilities were also encountered. Consequently, of the algorithms presented in this thesis, the only algorithm which successfully handles the case of correlated *a priori* is the Wiener filter.

We also find in practice that the Chandrasekhar filter is unstable when we attempt to perform adaptive filtering. We recall that one important objective was the smoothing of homogenous regions while recovering sharp transitions. In order to accomplish this task with the Chandrasekhar filter, we compute two sub-optimal gain vectors by using different values of μ. One gain vector is computed using a high value of μ which will smooth homogenous regions. The second is computed using a low value of μ which can be used to restore transitions. Then, the filtering can be performed by alternating between the two filters as a function of the local variance

of the image as described in Chapter 3. However, we find that the resulting algorithm is unstable. This is due to the fact that the Chandrasekhar filter presents an autoregressive structure so that at each iteration previous estimates strongly affect the current estimate. Consequently, when we change the gain vector in the middle of the filtering process, the quantity which is being updated at each recursion namely $\hat{x}$ becomes incompatible with the new vector gain.

By contrast with the constrained Wiener filter, the problem does not arise, since each pixel is being updated independently of the others.

4.3 Comparison of Wiener Filtering, Adaptive Wiener Filtering and Chandrasekhar Filtering on Simulated Images

The results for the Wiener filter were presented in Chapter 3. so we now turn our attention to a comparison of the various algorithms presented. In particular, we consider the benefits to be gained by using the Wiener filter instead of the Chandrasekhar filter. We are particularly interested in the improvement in performance associated with 3D filtering as well as adaptive filtering, both of which can be done by Wiener filtering.

For the sake of clarity in the graphs, we omit the optimal Chandrasekhar filter and use only the sub-optimal filter which uses the stationary gain. We recall however that the results with this filter are almost identical to those obtained with the optimal filter. In what follows, the term Chandrasekhar filter is used to refer to the sub-optimal filter.

Figure 4.12 contains a comparison of the sub-optimal Chandrasekhar filter, the 2D and the 3D Wiener filters. In performing the 3D filtering, we use 3 identical images with the seed of the random noise generator for the observation noise set differently for each one. This type of object is used since we observe that our real images of the knee vary very little from one slice to the next. The parameters for the Chandrasekhar filter are set empirically. The parameter λ for the non-adaptive 2D and 3D filters is also set empirically since as we recall from the article, problems are encountered if we use either a high or low value for λ determined by adjusting the value obtained from the CV curve. Automatic selection works well when we use the adaptive Wiener filter. The 3D filter yields a small improvement over the 2D Chandrasekhar filter and the 2D Wiener filter. We recall that no 3D implementation of the Chandrasekhar

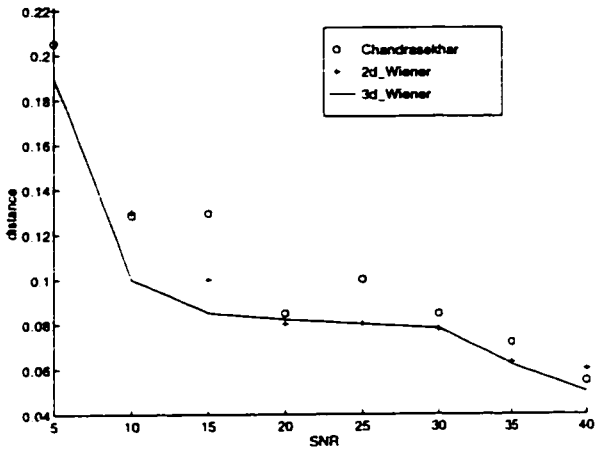


Figure 4.12: Comparison of Chandrasekhar, 2D and 3D non-adaptive Wiener filters

filter is possible due to the overwhelming computation time.

We now compare the results of the Chandrasekhar filtering with those of the adaptive Wiener filtering. In this case, the Chandrasekhar parameter μ is chosen empirically once again, and the parameters of the adaptive Wiener filter are chosen using CV as described in Section 3. The results comparing the Chandrasekhar with the 2D and the 3D adaptive Wiener filter are given in Figure 4.13. The 2D adaptive filter consistently outperforms the Chandrasekhar filter since errors in the homogenous zones can be substantially reduced by using a high parameter λ which corresponds to a smoothing filter. In the transition zones a lower value is used to recover the transition as best as possible. We note that the gain associated with using a 3D adaptive Wiener filter as opposed to a 2D adaptive Wiener filter is rather limited.

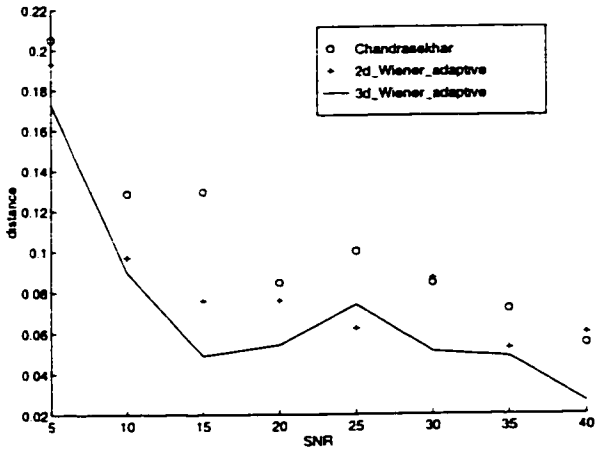


Figure 4.13: Comparison of Chandrasekhar, 2D and 3D adaptive Wiener filters

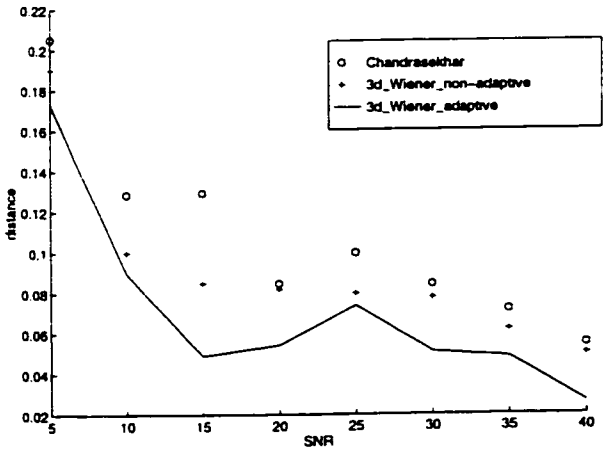


Figure 4.14: Comparison of Chandrasekhar, 3D non-adaptive and 3D adaptive Wiener filters

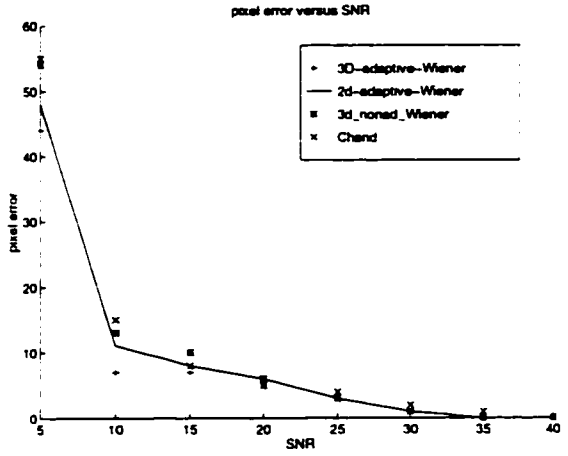
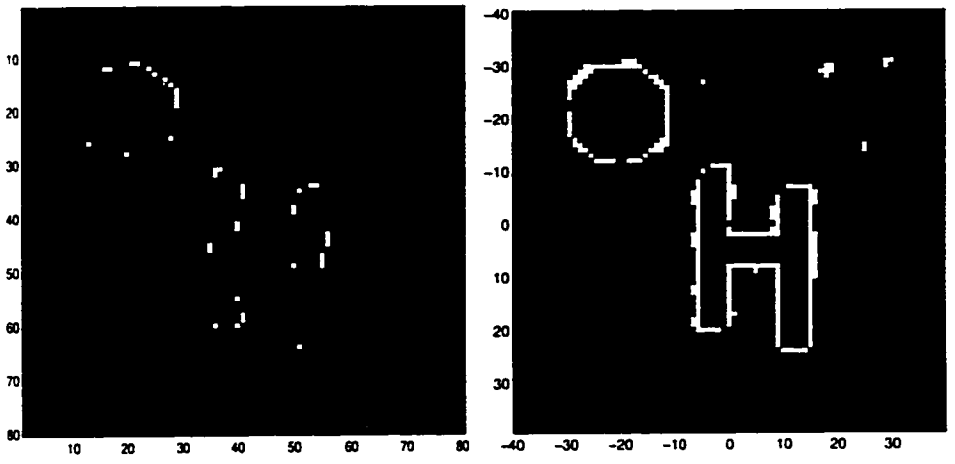


Figure 4.15: Pixel errors of Chandrasekhar and Wiener filters after threshold segmentation



(a) 3D adaptive Wiener filter (b) Chandrasekhar filter

Figure 4.16: Pixel errors after Segmentation; SNR=5dB

A comparison of the 3D adaptive Wiener filter, the 3D Wiener filter and the Chandrasekhar algorithm is given in Figure 4.14. We note that of the three filters, the 3D adaptive Wiener performs the best as measured by the square distance.

As was noted earlier, in practice we are interested in the segmented image as these will be used to obtain the contours. The results obtained after a segmentation by threshold are given in Figure 4.15. From this graph, we see that after segmentation, at higher SNRs (15dB or more) all the algorithms yield a very small pixel error. Below 15dB we see that the 3D adaptive Wiener filter yields a better result. This results from the fact that the adaptive filter allows us to smooth homogeneous regions so that pixels errors in these regions are eliminated. This is seen clearly when we look at the images of the pixel error as in Figures 4.16a and 4.16b obtained for an observed image at a SNR of 5dB.

Our results on simulated images indicate that after segmentation, the 3D adaptive Wiener filter performs best, though the improvement in going from 2D to 3D is limited. The improvement in performance is observed at SNRs under 15dB where the adaptive Wiener filter is able to smooth noise in homogeneous zones so as to reduce the pixel error. At higher SNRs the Chandrasekhar filter and the non-adaptive and adaptive Wiener filters, both in 2D and in 3D, all yield roughly the same result.

4.4 Results with real images

In this section we examine the performance of the algorithm on real images obtained from a tomographical scanner. Since the images are slices of the knee whose exact dimensions are unknown, it is impossible to quantitatively evaluate the performance of the algorithms. As such, we compare the performances of the filters qualitatively. We are particularly interested in the filter's ability to restore the edges and

suppress the noise. We find that qualitatively, of the filters developed, the constrained Wiener adaptive filter in both 2D and 3D perform better than the Chandrasekhar filter, particularly after a threshold segmentation.

Real images of the knee were obtained from a computerized tomography scanner. At first, the point spread function was measured by obtaining scans of fine metal wires immersed in water in a plastic drum (illustrated in Figure 4.17). The immersion in water is done for the purpose of more closely simulating body fluids. The wires were placed at various points in the scanner's field of view so as to measure the PSF at different locations. The image obtained is presented in Figure 4.18. From such an

Figure 4.17: Plastic drum with thin metal wires

image, the PSF is parameterized in terms of a Gaussian which closely approximates the observed PSF. When such a parameterization is made, we find that the the width of the PSF varies only by 10% to 20% (Doré and Goussard 1997). Results presented in Section 3 indicate that there is little to be gained in accounting for this variation. Consequently we use the central PSF illustrated in Figure 4.19.

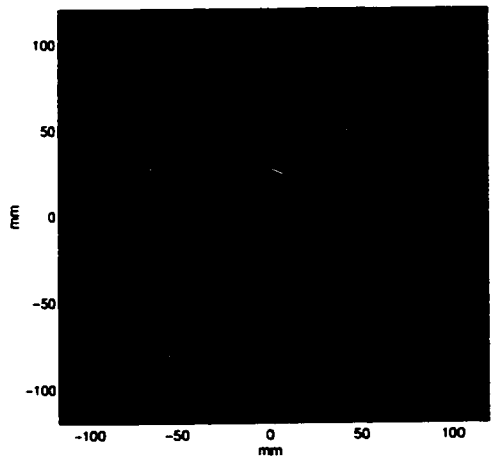


Figure 4.18: Measurements of the PSF at various locations

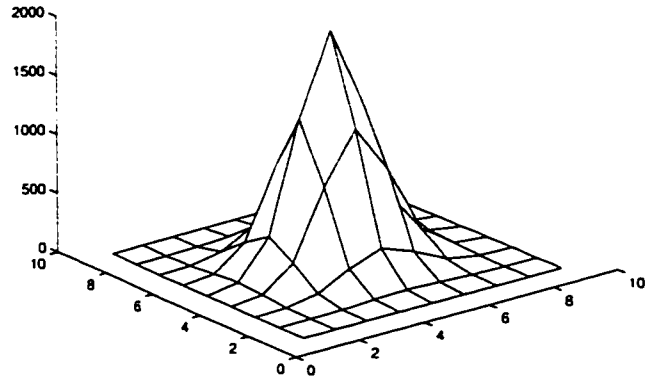
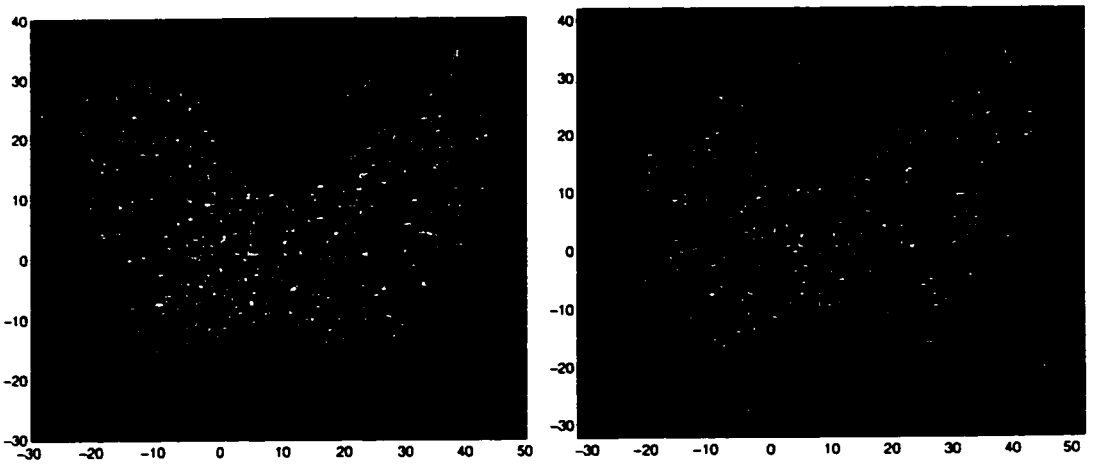


Figure 4.19: Central Point Spread Function

Then a series of images of the slices of the femur bone in the region of the knee were obtained. A real image of such a slice, obtained from the same tomographical scanner used to measure the PSF, is presented in Figure 4.20a. A close examination



(a) original

(b) restored with Chandrasekhar filter, $\mu = 0.2$

Figure 4.20: Slice of the femur obtained from a CT-scan restored by Chandrasekhar filter

of the image reveals the blurred nature of the edges. We also note that the noise present in the image is highly correlated which goes against our hypotheses. This image was restored using both the Chandrasekhar and the 2D Wiener filter. For the Chandrasekhar filter, the parameter μ is chosen empirically. The selected value of μ used in Chandrasekhar filtering was 0.2. In test images this value is used for restoration when the SNR is roughly 20dB. Consequently we can assume that the SNR in the real image is about 20dB. The resulting image appears in Figure 4.20b. We note that the blur has been removed, but noise is still present. As illustrated in Figure 4.21, if we raise the parameter μ in order to smooth the noise, we are unable to successfully remove the blur. In this image, it is clear that while we have smoothed the noise, the blur in the result is roughly the same as the original blur. Consequently,

we must compromise between removing the blur and suppressing the noise.

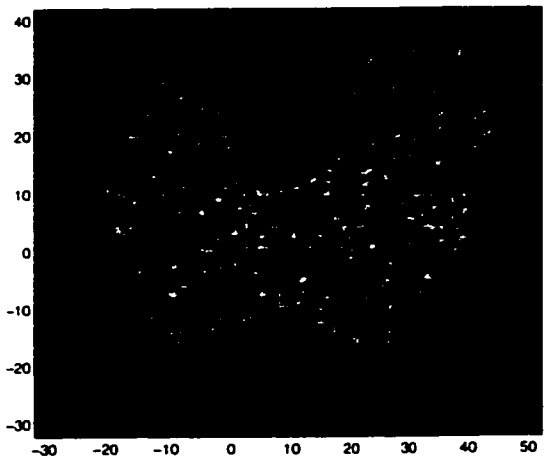
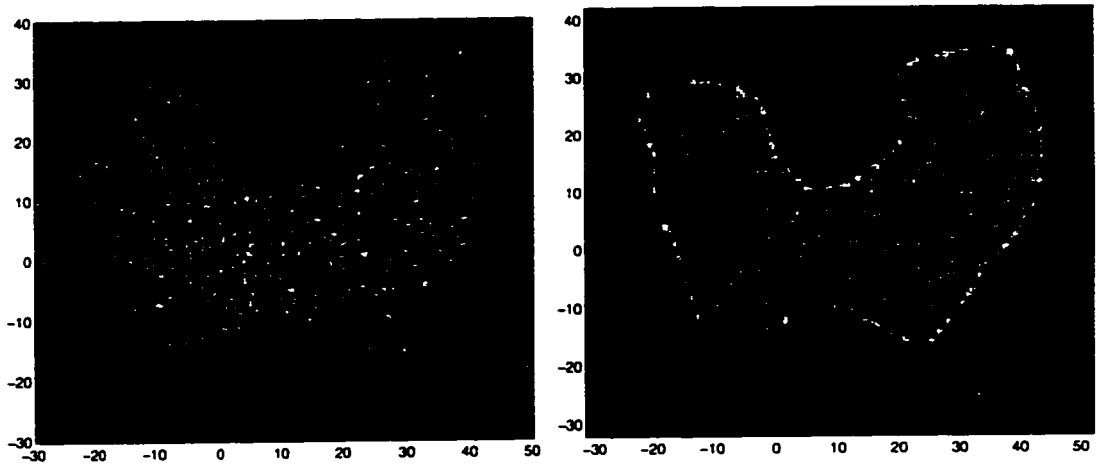


Figure 4.21: Restored image using Chandrasekhar filter with $\mu = 3$

For the case of the 2D adaptive Wiener filter, parameters are obtained using the CV algorithm presented in Section 3. Using such a filter yields the restored image in Figure 4.22b. The raw image is included in Figure 4.22a for comparison. The 2D adaptive Wiener filter allows us to suppress the noise in homogeneous regions while removing the blur in transitions by varying the parameters as a function of the local characteristics as described in Section 3.

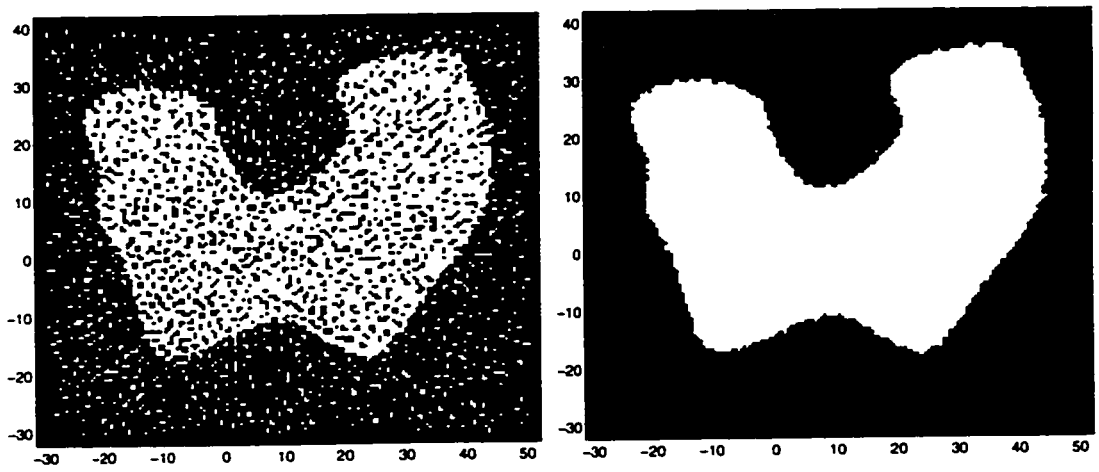
As discussed earlier, it is important to evaluate the performance after segmentation since it is from the segmented image that the contours will be extracted. We once again use a threshold as it is the simplest way to segment an image and furthermore it yielded good results on test images. Using a threshold to segment the images restored by the Chandrasekhar filter and the Wiener filter yields the images in Figures 4.23a and 4.23b. Clearly the smoothing performed by the adaptive filter helps yield a smoother segmented image. The segmentation of the image restored by Chandrasekhar filtering is inadequate.



(a) original

(b) restored by 2D adaptive Wiener filtering

Figure 4.22: Slice of the femur obtained from a CT-scan restored by 2D adaptive Wiener filter

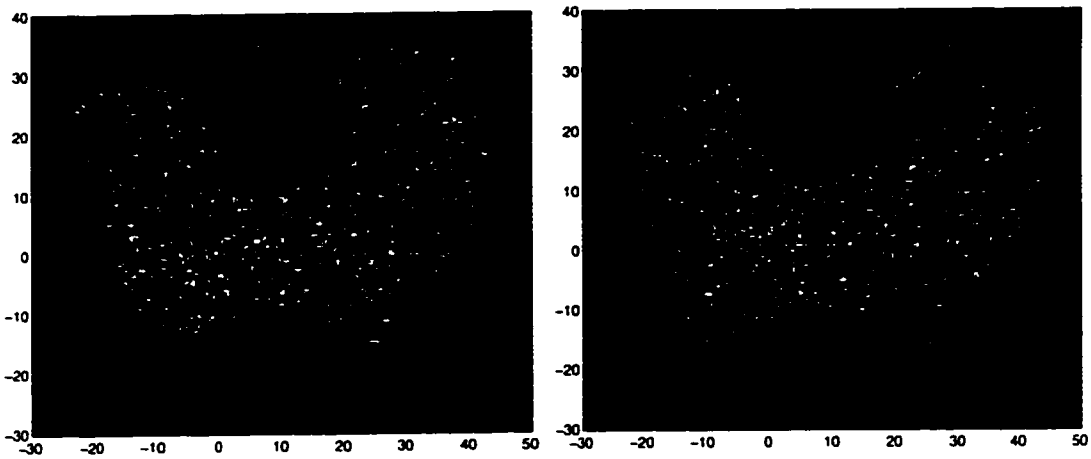


(a) Chandrasekhar

(b) 2d adaptive Wiener filtering

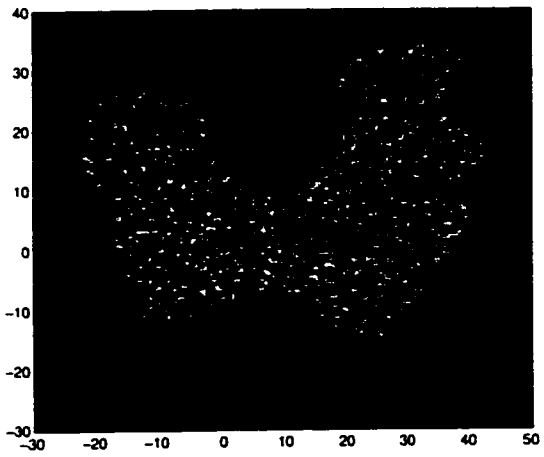
Figure 4.23: Results after segmentation by threshold

The 3D adaptive Wiener filter can also be used to restore the knee slice. In this case, the slices above and below the current slice are used for the filtering. The three slices used are illustrated in Figure 4.24. A 3D Wiener filter with a correlation between neighboring slices equal to 0.7 is used to restore the image. This yields the image in Figure 4.25. The image after segmentation by threshold is given in Figure 4.26. The result is very close to the segmentation obtained from the 2D wiener filter after a threshold is applied though the edges are rougher. We cannot conclude though that the 2D adaptive Wiener has performed better since we are unable to compute the pixel error. It is possible that the roughness is caused by the variation at the edges between adjacent slices. On test images, three identical images are considered so that we expect the restoration to be more accurate in this ideal case. It is possible in this case that the variation between slices caused certain pixels to be falsely classified. Once again though, we note that both the 2D and 3D adaptive filter perform markedly better than the Chandrasekhar filter which is unable to restore the edges while also smoothing the noise.



(a) lower slice

(b) middle slice



(c) upper slice

Figure 4.24: Knee slices used in the 3D restoration

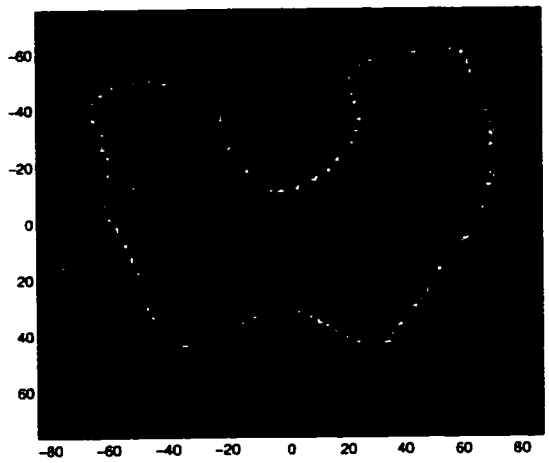


Figure 4.25: Image restored by 3D Wiener filter

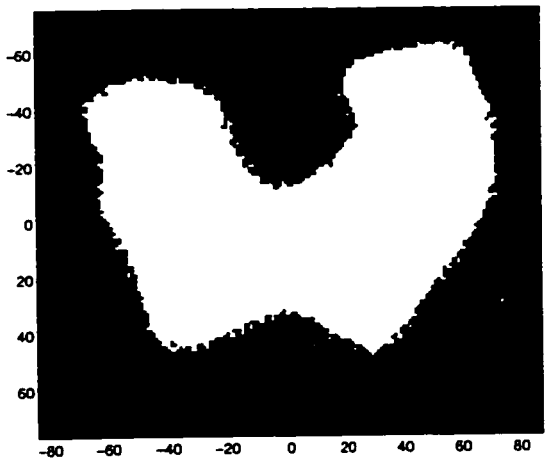


Figure 4.26: Segmented image after restoration by 3D Wiener filter

Since the true image of the slice of the knee is unknown, it is impossible to say how accurate the restoration process has been. In this respect, some work still needs to be done to calibrate the restoration algorithms. In particular, work has started in the fabrication of objects whose exact dimensions are known. Scans of these objects are made and the images can then be restored and the difference between the dimensions of the restored image and the dimensions of the object can be compared. However certain technical problems must be overcome. In particular, objects must be constructed so that corresponding points on the scanned image and the original object can be easily found. Consequently work is currently being done to construct objects with an embedded wire or groove that can be easily located on the scanned image.

We have seen in this chapter that of the filters developed, the 3D adaptive Wiener filter performs the restoration best on synthetic images as measured by the square error as well as the pixel error after segmentation. On real images, we cannot quantify the errors since the we do not have the original image. However, from a qualitative standpoint, it is clear that the Wiener adaptive filtering yields superior results. The parameter μ for the Chandrasekhar filter must be chosen either to restore the boundaries or to smooth the noise. By contrast the adaptive filter smooths the noise while restoring the transitions since the filter is tuned to the local characteristics of the image. Finally, as we have seen, using the adaptive filter to restore images, allows us to use a threshold segmentation. This type of segmentation is inadequate in the case where the image is restore by the Chandrasekhar filter.

Chapter 5

Conclusion

In this thesis we have looked at the problem of image restoration in the context of computer tomography images. Our ultimate goal was to reconstruct the surface of the knee in order to be able to fabricate a personalized knee prosthesis. Such a prosthesis requires a precision of roughly 1mm between the prosthesis and the surface of the knee. Consequently the imaging techniques used to reconstruct this geometry must be precise.

The algorithms presented in this thesis attempt to solve the first problem in the reconstruction process which is the problem of image restoration. We recall that our five objectives for the image restoration were the following: reduction of the blur, suppression of the noise, speed, automatic selection of parameters, and the ability to perform the filtering in 3D. Of these objectives, the most important objective was to obtain a suitable compromise between suppressing the noise and reducing the blur.

In the literature survey, we observed that non-quadratic criteria and Markov random fields in particular, allow the explicit introduction of homogeneous zones separated by transitions. However these methods lead to algorithms of rather high computational complexity. Furthermore, the problem of estimating parameters for non-

quadratic criteria has not been solved. Consequently we opted to pursue quadratic methods with the aim of avoiding the sub-optimalities associated with the reduced update Kalman filter. Furthermore we were interested in being able to filter the image based on the local characteristics and in particular the presence or absence of transitions.

The Kalman-based filtering methods developed attain only some of our objectives. We find that with these filters, these objectives are contradictory since attempting to overly remove the noise leads to a blurry image, while focusing on blur removal leads to a mediocre suppression of noise. Consequently a compromise must be made which is controlled by the parameters of the filter. This filter also suffers from the defect that no 3D extension is possible due to the overwhelming nature of the resulting calculations. However this class of filters is sufficiently fast to be used in practice, and automatic selection of the parameters is possible as illustrated in the survey of the literature.

In order to avoid the drawbacks of the Kalman filter, we developed another method based on the constrained Wiener filter. This method can be used adaptively so as to remove the blur around the edges while suppressing the noise in homogenous regions. Furthermore, the standard 2D filter lends itself easily to a 3D extension so we can incorporate a correlation between neighboring slices. Furthermore, while the Kalman based filters are unstable when correlation is introduced, the Wiener filter remains stable. The central original contribution of this thesis is the development of a parameter estimation algorithm based on cross validation which yields a reasonable approximation to the parameter λ which controls the degree of smoothing of the Wiener filter. With this algorithm, the 3D Wiener filter attains all the stated objectives.

A number of tests were conducted on simulated images. On these image we find

that the constrained 3D Wiener filter performs best. We find that the results are acceptable down to an SNR of 15dB. At 15dB, after a threshold segmentation we find an error of 1 pixel or less around the edges. This corresponds to an error of 0.5mm which is acceptable for our application.

The only real images on which the algorithm was tested were those of the knee obtained from a tomographical scanner. However, it is impossible to calibrate the method on these images as the true surface of the knee is unknown. We do note that qualitatively, the results appear good, since the blur is removed to a certain extent. Nevertheless, work remains to be done in the construction of objects whose exact dimensions are known. These objects must be constructed in such a way as to have reference points embedded in the object which can be located on the observed image and used to find a corresponding point on the true object. These objects must be placed in the scanner and images must be obtained. The resulting images can then be restored and the error between the restored image and the true object can be evaluated. However the scanner used yields images with an SNR of roughly 20dB and at these SNRs, we find that the Wiener filtering yields almost perfect restorations. Consequently, it is very likely that this type of filter can be used in practice.

Bibliography

ANDREWS, H. and HUNT, B. (1977). Digital image restoration. *Englewood Cliffs : Prentice-Hall.*

BANHAM, M. and KATSAGGELOS, A. (1997). Digital image resoration. *IEEE Signal Processing Magazine*, pp. 24–41.

BESNERAIS, G. L. and GOUSSARD, Y. (1993). Improves square-root forms of fast linear least squares estimation algorithms. *IEEE Transactions on Signal Processing*, *41*(3), 1412–1421.

BLAKE, A. (1989). Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-11*(1), 2–12.

BLAKE, A. and ZISSERMAN, A. (1987). *Visual Reconstruction.* MIT Press, Cambridge, MA.

BRETTE, S. and IDIER, J. (1996). Optimized single site update algorithms for image deblurring. *Proceedings of the International Conference on Image Processing*, Lausanne, Switzerland.

BROOKS, D. H., MARATOS, G. M. and MACLEOD, R. S. (1993). The augmented inverse problem of electrocardiography: Combined time and space regularization. *Proc. Annual IEEE-EMBS Conference*, pp. 773–774.

DACUNHA-CASTELLE, D. and DUFLO, M. (1982). *Probabilités et statistiques, 1. Problème à temps fixe.* Masson, Paris.

DEMOMENT, G. (1987). *Algorithme rapides.* Vol. 3152, Service d'Édition de SUPÉLEC. JFG.

DEMOMENT, G. (1989). Image reconstruction and restoration : Overview of common estimation structure and problems. *IEEE Transactions on Acoustics Speech and Signal Processing, ASSP-37*(12), 2024–2036.

DEMOMENT, G., REYNAUD, R. and SÉGALEN, A. (1983). Estimation sous-optimale rapide pour la déconvolution en temps-réel. *GRETSI9*, Nice, France, pp. 205–210.

DERIN, H. and KELLY, P. A. (1989). Discrete-index Markov-type random processes. *Proceedings of the IEEE, 77*, 1485–1510.

DORé, S. and GOUSSARD, Y. (1997). Experimental determination of CT point spread function anisotropy and shift-variance. Submitted to IEEE-EMBS Conf.

DURBIN, J. (1959). Efficient estimation of parameters in moving-average models. *Biometrika, 46*, 306–316.

EKSTROM M. P., WOODS, J. W. (1976). Two-dimensional spectral factorization with applications to recursive digital filtering. *IEEE Transactions on Acoustics Speech and Signal Processing, 24*, 115–124.

FORTIER, N., DEMOMENT, G. and GOUSSARD, Y. (1993a). Comparison of GCV and ML methods of determining parameters in image restoration by regularisation. *Journal of Visual Communication and Image Representation*, *4*, 157–170. JFG.

FORTIER, N., DEMOMENT, G. and GOUSSARD, Y. (1993b). GCV and ML methods of determining parameters in image restoration by regularization: Fast computation in the spatial domain and experimental comparison. *Journal of Visual Communication and Image Representation*, *4*(2), 157–170.

GALATSANOS, N. and KATSAGGELOS, A. (1992). Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation. *IEEE Transactions on Image Processing*, *IP-1*(3), 322–336.

GEMAN, S. and GEMAN, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-6*, 721–741.

GEMAN, S. and MCCLURE, D. (1987). Statistical methods for tomographic image reconstruction. *Proc. of the 46-th Session of the ISI, Bulletin of the ISI*, Vol. 52, pp. 22–26.

GEMAN, S. and REYNOLDS, G. (1992). Constrained restoration and recovery of discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-14*(3), 367–383.

GOLUB, G. and VAN LOAN, C. (1989). *Matrix computations (2nd Edition)*. John Hopkins University Press, London, England.

GOYETTE, J., LAIN, G. and KANG, M.G.AND KASTAGELLOS, A. (1994). Improving autoradiograph resolution using image restoration techniques. *IEEE Engineering Medicine Biology*, pp. 571–574.

GREEN, P. J. (1990). Bayesian reconstructions from emission tomography data using a modified EM algorithm. *IEEE Transactions on Medical Imaging*, *9*, 84–93.

GULL, S. F. (1984). *The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind.* Pittman, Boston, MA.

HUNT, B. (1977). Bayesian methods in nonlinear digital image restoration. *IEEE Transactions on Communications*, *C-26*, 219–229.

JENG, F.-C. and WOODS, J. W. (1990). Simulated annealing in compound Gaussian random fields. *IEEE Transactions on Information Theory*, *IT-36*(1), 94–107.

JENG, F.-C. and WOODS, J. W. (1991). Compound Gauss-Markov random fields for image estimation. *IEEE Transactions on Signal Processing*, *SP-39*(3), 683–697.

JOSEPH, P. M., SPITAL, R. D. and STOCKMAN, C. D. (1980). The effects of sampling on CT images. *Computerized Tomography*, *4*, 189–206.

KAILATH, T., VIEIRA, A. and MORF, M. (1978). Inverses of tœplitz operators, innovations, and orthogonals polynomials. *SIAM Review*, *20*, 106–119. JFG.

KIJEWSKI, M. F. and JUDY, P. F. (1983). The effects of misregistration of the projections on the spatial resolution of CT scanners. *Medical Physics*, *10*, 169–175.

LAGENDIJK, R., BIEMOND, J. and BOEKEE, D. E. (1989). Blur identification using the expectation minimization algorithm. *Proceedings of the IEEE*, pp. 1397–400.

LEVINSON, L. (1947). The wiener RMS criterion in filter design and prediction. *Journal of Math and Physics*, *25*, 261–278.

LI, F.-K., HELD, D. N., CURLANDER, H. C. and WU, C. (1985). Doppler parameter estimation for spaceborne synthetic-aperture radars. *IEEE Transactions on Geoscience and Remote Sensing*, *GE-23*(1), 47–56.

LI, K. C. (1986). Asymptotic optimality of c_l and generalized cross-validation in ridge regression with application to spline smoothing. *The Annals of Statistics*, *14*, 1101–1112.

LUENBERGER, D. (1969). *Optimization by Vector Space Methods.* 1 edn, Wiley, New York, NY.

MORF, M. and KAILATH, T. (1975). Square-root algorithms for least-squares estimation. *IEEE Transactions on Automatic and Control*, *AC-20*(4), 487–497.

NAGY, J. G., PLEMMONS, R. J. and TORGERSEN, T. C. (1996). Iterative image restoration using approximate inverse preconditioning. *IEEE Transactions on Image Processing*, *5*(7), 1151–1161.

NASHED, M. Z. (1981). Operator-theoretic and computational approaches to ill-posed problems with applications to antenna theory. *IEEE Transactions on Antennas and Propagation*, *AP-29*, 220–231.

PRESS, W., TEUKOLSKY, S., VETTERLING, W. and FLANNERY, B. (1992). *Numerical Recipes in C, the Art of Scientific Computing, second edition.* Cambridge Univ. Press, New York.

PROAKIS, J. G. and MANOLAKIS, D. G. (1988). *Digital Signal Processing: Principles, Algorithms and Applications.* MacMillan Publishing, New York, New York.

RADER, C. M. and STEINHARDT, A. O. (1986). Hyperbolic householder transformations. *IEEE Transactions on Acoustics Speech and Signal Processing*, *34*(6), 1589–1601.

RATHEE, S., ZOLY, J. K. and OVERTON, T. R. (1992a). Image restoration in computer tomography: Estimation of the spatially variant point spread function. *IEEE Transactions on Medical Imaging*, *11*(4), 539–545.

RATHEE, S., ZOLY, J. K. and OVERTON, T. R. (1992b). Image restoration in computer tomography: Restoration of experimental CT images. *IEEE Transactions on Medical Imaging*, *11*(4), 546–553.

RATHEE, S., ZOLY, J. K. and OVERTON, T. R. (1992c). Image restoration in computer tomography: The spatially invariant point spread function. *IEEE Transactions on Medical Imaging*, *11*(4), 530–538.

REICHENBACH, S. E. and PARK, S. K. (1991). Small convolution kernels for high-fidelity image restoration. *IEEE Transactions on Signal Processing*, *39*(10), 2263–2273.

REICHENBACH, S. E., KOEHLER, D. E. and STRELOW, D. W. (1995). Restoration and reconstruction of AVHRR images. *IEEE Transactions on Geoscience and Remote Sensing*, *33*(4), 997–1007.

SAUER, K. and BOUMAN, C. (1993). A local update strategy for iterative reconstruction from projections. *IEEE Transactions on Signal Processing*, *SP-41*(2), 534–548.

SEZAN, M. I. and STARK, H. (1982). Image restoration by the method of convex projections : Part 2 - applications and numerical results. *IEEE Transactions on Medical Imaging, MI-1*(2), 95–101.

SOLTANIAN-ZODIAH, H., WINDHAM, J. and YAGLE, A. (1995). A multidimensional nonlinear edge-preserving filter for magnetic resonance image restoration. *IEEE Transactions on Image Processing, 4*, 147–161.

STEINHARDT, A. O. (1988). Householder transforms in signal processing. *IEEE ASSP Magazine, 53*, 4–12.

STEVENSON, R. and DELP, E. (1990). Fitting curves with discontinuities. *Proc. of the First International Workshop on Robust Computer Vision*, pp. 127–136.

THOMPSON, A., BROWN, J. C., KAY, J. W. and TITTERINGTON, D. M. (1991). A study of methods of choosing the smoothing parameter in image restoration by regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-13*(4), 326–339.

VERHAEGEN, M. and VAN DOOREN, P. (1986). Numerical aspects of different kalman filter implementations. *J. of the American Statistical Association, Theory and Methods Section, AC-31*(10), 907–917.

VERLY, J. G. and BRACEWELL, R. N. (1979). Blurring in tomograms made with X-ray beams of finite width. *JCAT, 3*, 662–678.

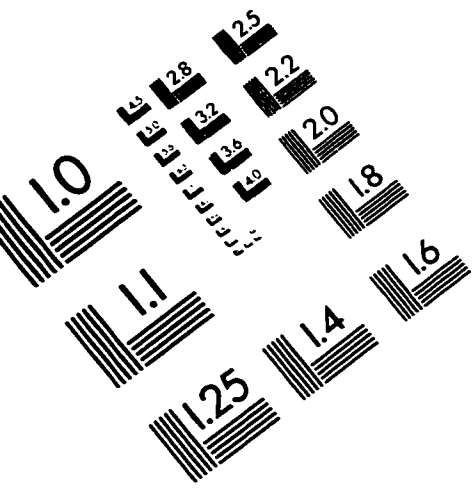
WAHBA, G. (1985). A comparison of GCV and GML for choosing the smoothing paramteted in the generalized spline smoothing problem. *Annals of Statistics, 13*, 1378–1402.

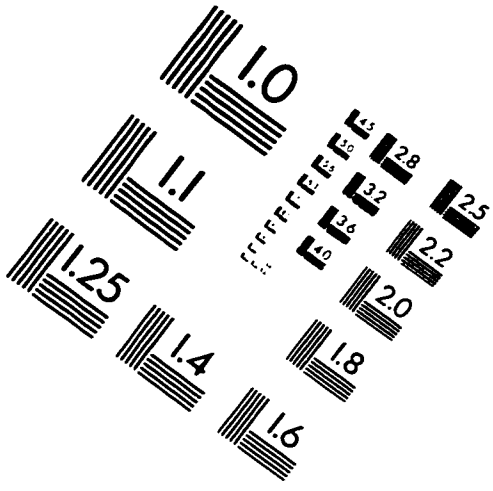
WOODS, J. W. and EKSTROM, M. P. (1975). Nonsymmetric half-plane recursive filter:characterization, stability theory and test. *Proceedings 1975 Int'l Sympos. on Circuits and Systems*, pp. 447–450.

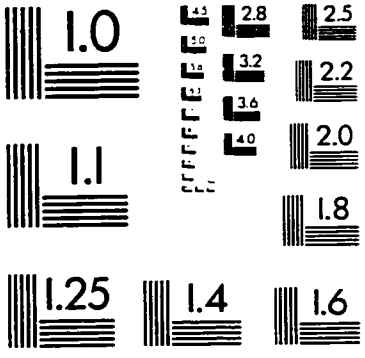
WOODS, J. W. and INGLE, V. K. (1981). Kalman filtering in two dimensions:further results. *IEEE Transactions on Acoustics Speech and Signal Processing*, *29*(2), 188–197.

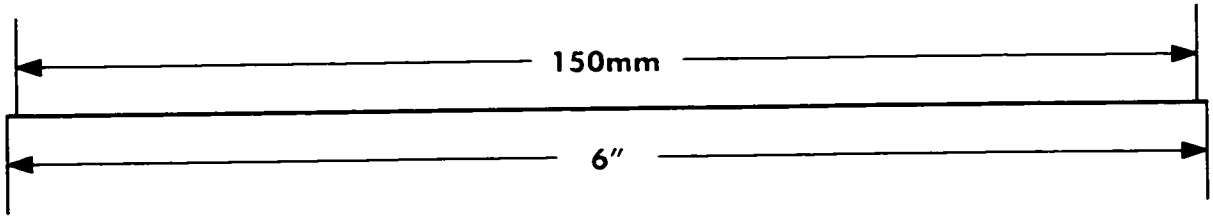
WOODS, J. W. and RADEWAN, C. H. (1977). Kalman filtering in two dimensions. *IEEE Transactions on Information Theory*, *23*(4), 473–482.

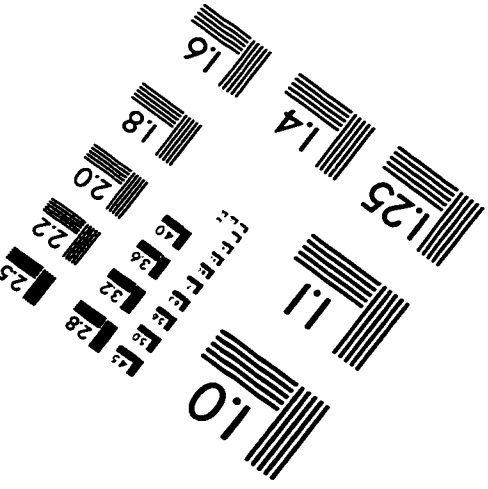
IMAGE EVALUATION
TEST TARGET (QA-3)

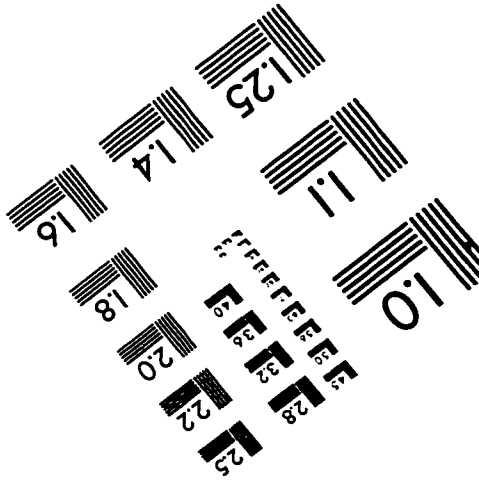












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989